# Convex Optimization and Extensions, with a View Toward Large-Scale Problems

Wenbo Gao

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

**COLUMBIA UNIVERSITY**

2020

# Abstract

## Convex Optimization and Extensions, with a View Toward Large-Scale Problems

Wenbo Gao

Machine learning is a major source of interesting optimization problems of current interest. These problems tend to be challenging because of their enormous scale, which makes it difficult to apply traditional optimization algorithms. We explore three avenues to designing algorithms suited to handling these challenges, with a view toward large-scale ML tasks. The first is to develop better general methods for unconstrained minimization. The second is to tailor methods to the features of modern systems, namely the availability of distributed computing. The third is to use specialized algorithms to exploit specific problem structure.

Chapters 2 and 3 focus on improving quasi-Newton methods, a mainstay of unconstrained optimization. In Chapter 2, we analyze an extension of quasi-Newton methods wherein we use block updates, which add curvature information to the Hessian approximation on a higher-dimensional subspace. This defines a family of methods, Block BFGS, that form a spectrum between the classical BFGS method and Newton's method, in terms of the amount of curvature information used. We show that by adding a correction step, the Block BFGS method inherits the convergence guarantees of BFGS for deterministic problems, most notably a $Q$-superlinear convergence rate for strongly convex problems. To explore the tradeoff between

reduced iterations and greater work per iteration of block methods, we present a set of numerical experiments.

In Chapter 3, we focus on the problem of step size determination. To obviate the need for line searches, and for pre-computing fixed step sizes, we derive an analytic step size, which we call curvature-adaptive, for self-concordant functions. This adaptive step size allows us to generalize the damped Newton method of Nesterov to other iterative methods, including gradient descent and quasi-Newton methods. We provide simple proofs of convergence, including superlinear convergence for adaptive BFGS, allowing us to obtain superlinear convergence without line searches.

In Chapter 4, we move from general algorithms to hardware-influenced algorithms. We consider a form of distributed stochastic gradient descent that we call Leader SGD, which is inspired by the Elastic Averaging SGD method. These methods are intended for distributed settings where communication between machines may be expensive, making it important to set their consensus mechanism. We show that LSGD avoids an issue with spurious stationary points that affects EASGD, and provide a convergence analysis of LSGD. In the stochastic strongly convex setting, LSGD converges at the rate $O(\frac{1}{k})$ with diminishing step sizes, matching other distributed methods. We also analyze the impact of varying communication delays, stochasticity in the selection of the leader points, and under what conditions LSGD may produce better search directions than the gradient alone.

In Chapter 5, we switch again to focus on algorithms to exploit problem structure. Specifically, we consider problems where variables satisfy multiaffine constraints, which motivates us to apply the Alternating Direction Method of Multipliers (ADMM). Problems that can be formulated with such a structure include representation learning (e.g with dictionaries) and deep learning. We show that ADMM can be applied directly to multiaffine problems. By extending the theory of nonconvex ADMM, we prove that ADMM is convergent on multiaffine problems satisfying certain assumptions, and more broadly, analyze the theoretical properties of ADMM for general problems, investigating the effect of different types of structure.

# Table of Contents

# List of Tables

# List of Figures

## Acknowledgements

First, my deep gratitude to my advisor, Don Goldfarb. This journey would not have been possible without his guidance, in research and in life. Don's work, which has shaped modern optimization, is an inspiration to me, and I can only hope to be as prolific. I am honored to be his student.

I thank my other collaborators: Frank E. Curtis and Anna Choromanska, for interesting problems and excellent papers;

Krzysztof Choromanski, for introducing me, with his usual inexhaustible energy, to exciting new areas of reinforcement learning;

Xingyou Song, Laura Graesser, and Vikas Sindhwani, for fruitful collaborations and great times during my Google internship.

I thank Dan Bienstock and Shipra Agrawal, for serving on my committee, and for the inspiration of their work.

I thank the department staff for handling administrative matters. In particular, thanks to Liz Morales for going above and beyond.

I thank my friends; a better group you could not find. In no particular order: RR, RX, AC, RJ, MO, FL, CZ, ZQ, EB, FF, SY, GXY, JL, MU, JZ, AZ, KG, MH, RS, SY, YT, MX, BY, and AL.

Finally, my gratitude to my parents, Min Li and Luomin Gao, for supporting me throughout my life. Nothing would have been possible without you.

*To my parents.*

# Preface

The main chapters of this thesis are closely adapted from published papers:

- Chapter 2 is based on the article *Block BFGS Methods* [1], SIAM Journal on Optimization 28(2), 2018. This article is joint with Donald Goldfarb.

- Chapter 3 is based on *Quasi-Newton Methods: Superlinear Convergence without Line Searches for Self-Concordant Functions* [2], Optimization Methods and Software 34(1), 2018. This article is joint with Donald Goldfarb.

- Chapter 4 is based on *Leader Stochastic Gradient Descent for Distributed Training of Deep Learning Models* [3], Advances in Neural Information Processing Systems 32, 2019. This article is joint with Yunfei Teng, Francois Chalus, Donald Goldfarb, Anna Choromanska, and Adrian Weller. The results in Chapter 4 are the work of the author.

- Chapter 5 is based on *ADMM for Multiaffine Constrained Optimization* [4], Optimization Methods and Software 35(2), 2020. This article is joint with Donald Goldfarb and Frank E. Curtis.

# Chapter 1: Introduction

The field of optimization has progressed enormously in the past century. The advent of electronic computers, and the subsequent rapid growth in computing power, made mathematical modeling and optimization applicable to a vast number of new problems. Though modern methods are based on the same fundamental principles that date back to the invention of calculus, many new challenges have arisen as the scale and complexity of optimization problems grows, with ever more ambitious problems entering the realm of tractability over time. This calls for new advances in optimization methods, in parallel with improvements in computer hardware.

The predecessors of modern optimization methods have a long history, and arose in the context of solving systems of equations in physics, rather than optimization per se. *Gradient descent*, which is now ubiquitous and perhaps the most fundamental method, was described (in an early form) by Cauchy in his 1847 paper [5] on computing orbits of astronomical bodies. The method now known as *Newton's method*, or the *Newton-Raphson method*, was developed by Viète [6], Newton [7, 8] and Raphson [9] as an algorithm for solving nonlinear systems of equations, motivated again by astronomy (in Newton's case). Simpson [10] was perhaps the first to explicitly identify that Newton's method could be used for function maximization by solving the system $f'(x) = 0$. A detailed survey of the history of Newton's method is available in [11].

Both gradient descent and Newton's method continue to be powerful and useful methods. However, using the basic version of these algorithms on modern problems is often impractical. In recent years, many optimization problems of interest have originated in *machine learning*. Without delving deeply into the statistical origins of machine learning, which is beyond the scope of this thesis, these problems can often be formulated as:

$$\min_x f(x) = \mathbb{E}_\xi \ell(x; \xi)$$

where $x \in \mathbb{R}^n$ is the decision variable, $\xi$ is a random variable that typically represents different members of an underlying population, and $\ell(x;\xi)$ is a loss function measuring the performance of the parameters $x$ on the instance $\xi$. In practice, the population distribution of $\xi$ is not available, and we instead perform *empirical risk minimization* (ERM), minimizing the empirical loss:

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{N}\sum_{i=1}^{N} \ell(x;\xi_i)$$

where $\{\xi_1, \ldots, \xi_N\}$ is a set of data points.

The number of parameters ($n$) is often extremely large. Indeed, the most successful class of machine learning algorithms, *deep learning*, make use of heavily over-parameterized models. Breakthroughs in computer vision [12] and natural language processing [13] have used increasingly large models; the Resnet-1202 network [12] has 19.4 million parameters, and BERT-LARGE [13] has 340 million parameters. Recent experiments show that deep learning often exhibits a 'double descent' curve [14], which defies the classical tradeoff between bias and variance. Instead, a new regime exists when the number of model parameters exceeds the 'interpolation threshold'; below the threshold, generalization performance follows the classical U-shaped curve, but then descends again as the model size grows further. Practitioners often favor larger models, especially in natural language processing, where state-of-the-art models may have sizes in the range of one to ten *billion* parameters (e.g. [15]).

Another challenge arises because the number of data points $N$ is often large, and indeed, must be for machine learning to be effective. This makes it prohibitively time-consuming to evaluate the full average $\frac{1}{N}\sum_{i=1}^{N} \ell(x;\xi_i)$ over all $N$ points, or the average of the gradients. Instead, it is standard to use a *stochastic* method, which at each iteration subsamples a *minibatch* of the data points and estimates the function or gradient over the minibatch. This algorithm, *stochastic gradient descent* (SGD), is the underlying tool that enables machine learning [16, 17]. The use of minibatches opens up new avenues, from exploiting hardware to most efficiently parallelize over data, to adapting algorithms to mitigate the additional stochasticity from subsampling.

The problems arising from machine learning often have several features which makes it difficult to directly apply classical optimization algorithms.

1. When the number of parmameters is large, it is impossible to use any implementation of Newton's method which stores a dense Hessian matrix. A model with $10^8$ parameters has roughly $\frac{1}{2} \cdot 10^{16}$ entries[1] which requires 10 *quadrillion* bytes of memory if each entry is stored as a half-precision floating point number.

2. Loss functions induced by ERM are often *highly* nonconvex, and even when convex, can be very ill-conditioned. Gradient descent is known to converge extremely slowly on such problems.

In this thesis, we focus on three different approaches to enhancing and extending optimization methods to handle the challenges of large-scale problems. We first introduce the approaches here and provide a brief summary. In the following sections, we describe each approach in greater detail, which also serves as an overview of the chapters of this thesis.

**Enhanced General Algorithms** Gradient descent, BFGS, and Newton's method are all instances of *general* methods for unconstrained optimization problems. That is, they apply to any problem of the form

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f$ is assumed only to be smooth to a sufficient order. Under certain conditions, such as (strong) convexity, it can be shown that these methods are convergent, and at a particular rate. In Section 1.1, we focus on *quasi-Newton methods*, and describe techniques for *better general-purpose quasi-Newton* algorithms. The two main aspects we consider are *increasing the use of curvature* and *better step size selection*.

**Distributed Computing** The performance of sequential algorithms is inherently bounded by physical limitations on processor speed, which may be approaching [18]. To surpass these limits,

[1] Assuming we store only the upper triangle of the Hessian.

there has been an increased focus on the use of *parallel and distributed computing* to make use of widely available computers, which individually may not be very powerful. The strategy of using distributed commodity hardware instead of expensive specialized systems is often efficient and more practical [19]. However, parallel and distributed algorithms are also inherently more complicated than sequential algorithms, with the new mechanic of communication. While there is a straightforward way to parallelize Stochastic Gradient Descent for machine learning using the *data-parallel* paradigm, this technique incurs high communication costs which is ill-suited for computers distributed over a network. In Section 1.2, we discuss a new *distributed SGD* method with multiple independent copies of the parameters, allowing for reduced communication costs.

**Structured Problems** Problems often belong to classes which have additional structure beyond the general problem $\min f(x)$. An important example is *linear programming* (LP), where the objective and constraints consist of linear functions. Extremely efficient specialized algorithms exist for LP which take advantage of the linear structure. The class of ERM problems can also be considered as a structured class, with algorithms specifically designed to exploit the *finite-sum* nature of the objective function to obtain speedups over batch gradient descent [20, 21, 22].

In Section 1.3, we consider a different class of structured problems, namely those with a *separable* structure amenable to the *alternating direction method of multipliers* (ADMM). Our interest is in generalizing ADMM to problems where variables are coupled in a *multiaffine* fashion, which arises when learning representations from data.

## 1.1 Improving General Optimization Methods

General optimization methods are those which do not rely on any special properties of the problem to be solved. As mentioned above, two of the most important examples, which are closely linked with the history of optimization, are *gradient descent* and *Newton's method*.

These methods can be viewed as part of a larger spectrum. Gradient descent is a *first-order method* in that it assumes the objective function to be differentiable, and uses only the value of the objective function (the zero-th order) and the gradient. Newton's method is a *second-order method* which assumes the objective function to be twice differentiable, and uses both the gradient and the Hessian. Neither method is unambiguously superior to the other across all problems. A trade-off exists between the convergence rate and the computational expense per step. While gradient descent can achieve only a sublinear or linear rate of convergence, each step is typically fast to compute, whereas Newton's method can achieve (local) quadratic convergence at the expense of *computing the Hessian* and *solving the Newton system*.

*Quasi-Newton* (QN) methods exist between these two extremes. A QN method maintains an approximation $B_k$ of the Hessian and generates steps by solving the Newton system using its Hessian approximation, i.e

$$x_{k+1} = x_k - \lambda_k B_k^{-1} \nabla f(x_k)$$

The most successful quasi-Newton method is *BFGS* [23, 24, 25, 26], which uses a particular updating scheme for the matrices $B_k$. BFGS itself is part of a spectrum of QN methods known as the *Broyden class* [27], which includes BFGS as one endpoint and the *DFP* method [28, 29] as the other[2]. The methods of the Broyden class aim to incorporate information about the true Hessian action along *one dimension* into the approximation $B_k$, using a rank-two update matrix.

There are two main ingredients to improving quasi-Newton methods:

- How can we best make use of curvature information in the matrix $B_k$?
- How do we select the step sizes $\lambda_k$?

### 1.1.1 Curvature in Quasi-Newton methods

The idea of increasing the amount of curvature information stored in the Hessian approximation dates back to Schnabel [30], who considered systems of secant equations. However, this approach

[2]The DFP method is the first known quasi-Newton method.

had technical limitations and received little attention until it was revisited for machine learning problems in [31]. A major point of divergence in [31], which overcame the technical issues of [30], was the use of the true Hessian action $\nabla^2 f(x_k) \cdot s_k$ on the vector $s_k$ as opposed to the gradient difference $\nabla f(x_k + s_k) - \nabla f(x_k)$. This was proposed in an earlier work on stochastic L-BFGS for machine learning [32], who noted that using a subsampled stochastic Hessian action produced better results than differencing subsampled stochastic gradients. It was also noted that in many machine learning problems, the (subsampled) Hessian-vector product $\widetilde{\nabla}^2 f(x_k) \cdot d_k$ could be computed in roughly the same time as the subsampled gradient, making it practical to use even for problems with a large number of parameters. This led to the *Stochastic Block L-BFGS* method of [31], which proposed performing BFGS updates with higher-dimensional systems $B_k D_k = \widetilde{\nabla}^2 f(x_k) D_k$, where $D_k = [d_1 \dots d_q]$ is a column matrix of $q$ directions.

This leads to a new spectrum of QN methods, which we call *Block BFGS*, with varying amounts of curvature information, depending on the number $q$ of directions in the update. For $q = 1$, we recover a rank-two update similar to the classical BFGS method, whereas $q = n$ is equivalent to Newton's method. Intermediate values of $q$ allow us to trade off between using more second-order information, and having to compute more Hessian-vector products.

While the $q = n$ case is generally equivalent to Newton's method (it holds if the Hessian and $D_k$ are both nonsingular), the $q = 1$ case is not equivalent to BFGS. In the original BFGS method, the gradient difference $\nabla f(x_k + d_k) - \nabla f(x_k)$ is guaranteed to have certain useful properties because of the Armijo-Wolfe line search used to find $\lambda_k$. Replacing it by the Hessian action $\nabla^2 f(x_k) \cdot d_k$ does not guarantee the same properties hold. This leaves unanswered the question of whether Block BFGS, even with $q = 1$, is convergent, and at what rates.

This is the topic addressed in Chapter 2. We show that with a minor modification to detect degeneracy of $D_k$, the Block BFGS method is globally convergent on convex problems, and recovers the $Q$-superlinear convergence of BFGS for strongly convex problems. We also perform several experiments to compare different methods in the Block BFGS family, and explore the tradeoff between using more curvature, and computational time.

### 1.1.2 Step Sizes for Self-Concordant Functions

Selecting the step size $\lambda_k$ is crucial for good performance. Two common strategies are *line searches* and *constant step sizes*. A *line search* is an algorithm which returns a step $\lambda_k$ guaranteed to satisfy certain conditions; a commonly used type is *Armijo-Wolfe line search* which has two conditions (2.2.1) and (2.2.2). In contrast, the constant step size strategy selects a single $\lambda$ and fixes $\lambda_k = \lambda$ for all steps.

Both strategies have advantages and drawbacks. Line search is adaptive to the local region of the objective function, and can take larger steps to speed up convergence when appropriate. However, it requires additional computation at each step, including multiple evaluations of the objective function and its gradient at various candidate points. For large-scale problems in machine learning (large $N$), even the evaluations of the loss function are generally too costly, making line search impractical.

The constant step size approach requires no additional computation during the running of the algorithm, but instead offloads the effort to the meta-problem of setting the hyperparameter $\lambda$. Moreover, using a constant $\lambda$ may be inefficient, since the iterates may move between regions of high and low curvature, for which vastly different step sizes are appropriate. This may result in convergence speed being degraded.

Our goal is to find an *analytic* step size which is both computationally efficient and has useful convergence guarantees. This is the question addressed in Chapter 3. We take a cue from the *damped Newton method* of Nesterov [33], which is a *globally convergent* Newton method for the class of *self-concordant functions*. Note that the original Newton method uses $\lambda_k = 1$ and is only *locally* convergent, even for strongly convex functions [34]. We extend the step size of the damped Newton method to a *curvature-adaptive step size* which can be applied to any iterative optimization method, including gradient descent and BFGS in particular. Our curvature-adaptive step size has a simple analytic expression, and requires only a single Hessian-vector product to evaluate. We show that gradient descent and BFGS retain their convergence guarantees with this step size, and compare it against other schemes in both the deterministic and stochastic settings.

## 1.2 Making use of Distributed Computing

Making effective use of hardware is essential for good performance. Part of the recent success of machine learning can be attributed to the rise of hardware acceleration, such as using GPUs [35, 36]. Though originally designed for other applications, these accelerators are optimized for fast numerical linear algebra, which is also the core operation of deep learning.

The same algorithm may be implemented in different ways, with dramatic differences in real computing time. Consider for example the basic SGD algorithm, which uses the gradient of the loss function on a random minibatch of size $m$. The gradient $\nabla \ell(x; \xi_i)$ for a particular sample $\xi_i$ is independent of that for any other sample $\xi_j$, and hence can be parallelized. For systems which support it, computing the gradients of the samples in parallel yields an almost $m$-fold speedup over sequentially computing the gradients. On a device such as a GPU, parallelism exists at multiple levels, for both speeding up primitive operations such as matrix multiplication, and for parallelizing over data samples.

This strategy of parallelizing calculations over samples is known as *data parallelism*, and is supported by almost all major deep learning frameworks [36]. However, its scalability in terms of the number of machines is ultimately limited by factors such as the communication cost, and the reduced generalization of large minibatches, which must be mitigated by other techniques [37]. Instead, we may consider algorithms which allow for different machines (or *workers*) to keep independent copies of the model parameters, and perform local training. Our starting point for such algorithms is *Elastic Averaging SGD* (EASGD) [38], which solves a *global variable consensus* problem:

$$\min_{x^{(1)},\ldots,x^{(p)},\widetilde{x}} \frac{1}{p}\sum_{i=1}^{p} \mathbb{E}_{\xi}\ell(x^{(i)};\xi) + \frac{\rho}{2}\|x^{(i)} - \widetilde{x}\|^2. \tag{1.2.1}$$

Each of the $p$ workers has its own set of the parameters $x^{(i)}$, and are coupled by the penalty function $\|x^{(i)} - \widetilde{x}\|^2$. A central machine (or *coordinator*) maintains $\widetilde{x}$, the *consensus variable*, which is responsible for maintaining consistency between the workers.

The EASGD method is powerful, but has several disadvantages in theory and practice which

arise from its design. The consensus variable $\widetilde{x}$ is a decision variable, which causes the global objective function (1.2.1) to have spurious stationary points. This is especially apparent when the underlying loss function $f$ has many symmetries, and the workers are attracted to different local minima.

In Chapter 4, we consider an alternate scheme called *Leader SGD* (LSGD), where the consensus variable is no longer a decision variable, but rather is computed from the worker $x^{(1)}, \ldots, x^{(p)}$ by estimating the best parameters. We show that LSGD avoids certain pitfalls of EASGD, and analyze the convergence of EASGD. We show that EASGD matches the convergence rate of other distributed SGD algorithms, and study its properties under various levels of stochasticity and communication delay.

### 1.3 Algorithms for Structured Problems

We are interested in problems which are amenable to ADMM. A motivating example is the separable problem having the form $\min_{x\in\mathbb{R}^n} f(x)+g(x)$ where $f, g$ are individually straightforward to minimize, but their sum is not. An instance which arises often in machine learning is the use of regularization; for instance, we may have

$$\min_{x\in\mathbb{R}^n} \frac{1}{N}\sum_{i=1}^{N} \ell(x;\xi) + \|x\|_p.$$

This type of problem can be solved with the ADMM method, which is specialized for separable objective functions with linearly constrained variables:

$$\begin{cases} \min\limits_{x,y\in\mathbb{R}^n} & f(x)+g(y) \\ s.t. & x = y \end{cases}$$

A comprehensive survey of ADMM and its applications can be found in [39].

Existing work on ADMM has focused on the case of linear constraints. However, variables are often coupled in more complex ways. When learning representations of data, it is often neces-

sary to learn both the set of *representatives* as well as the actual representation of the given data corresponding. This is typically a bilinear operation, such as convolution. For example, noisy observations $B$ can be modeled as the result of applying a convolution $A$ to an underlying signal $X$, and we aim to recover both the convolutional kernel and the signal given access to the observation. The variables then satisfy the relation $A * X = Y$, where $*$ denotes a convolution. While this constraint alone makes it ill-posed to recover $A$ and $X$, sparsity assumptions on $X$ make this a well-defined optimization problem. Observe that when $A$ is fixed, the resulting equation becomes *linear* in $X$, and likewise for $A$ when $X$ is fixed. This suggests that ADMM may be applicable to solving this problem.

In Chapter 5, we investigate the properties of ADMM for problems where the constraints are *multiaffine*. For such problems, ADMM can be applied in the same way as for linearly-constrained problems, since the subproblems have the same structure when minimizing for each variable in turn. We show that under similar assumptions as those used for the analysis of linearly-constrained nonconvex ADMM, we obtain convergence of ADMM when applied to multiaffine problems, and present examples of problems with a multiaffine structure.

# Chapter 2: Block BFGS Methods

## 2.1 Introduction

The classical BFGS method [23, 24, 26, 25] is perhaps the best known *quasi-Newton method* for minimizing an unconstrained function $f(x)$. These methods iteratively proceed along search directions $d_k = -B_k^{-1}\nabla f(x_k)$, where $B_k$ is an approximation to the Hessian $\nabla^2 f(x_k)$ at the current iterate $x_k$. Quasi-Newton methods differ primarily in the manner in which they update the approximation $B_k$. The BFGS method constructs an update $B_{k+1}$ that is the nearest matrix to $B_k$ (in a variable metric) satisfying the *secant equation* $B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)$ [23]. This can be interpreted as modifying $B_k$ to act like $\nabla^2 f(x)$ along the step $x_{k+1} - x_k$, so that successive updates induce $B_k$ to resemble $\nabla^2 f(x)$ along the search directions.

A natural extension of the classical BFGS method is to incorporate information about $\nabla^2 f(x)$ along *multiple* directions in each update. This further improves the accuracy of the local Hessian approximation, allowing one to obtain better search directions. Early work in this area includes the development by Schnabel [30] of quasi-Newton methods that satisfy multiple (say, $q$) secant equations $B_{k+1}s_k^{(i)} = \nabla f(x_{k+1}) - \nabla f(x_{k+1} - s_k^{(i)})$ for directions $s_k^{(1)}, \ldots, s_k^{(q)}$. This approach has the disadvantage that the resulting update is generally not symmetric, and considerable modifications are required to ensure $B_k$ remains positive definite. Consequently, there appears to have been little interest in quasi-Newton methods with block updates in the years following Schnabel's initial report.

More recently, a stochastic quasi-Newton method with block updates was introduced by Gower, Goldfarb, and Richtárik [31]. Their approach constructs an update which satisfies *sketching equations* of the form

$$B_{k+1}s_k^{(i)} = \nabla^2 f(x_{k+1})s_k^{(i)}$$

for multiple directions $s_k^{(i)}$. By using sketching equations instead of secant equations, the update is guaranteed to remain symmetric, and in the case where $f(x)$ is convex, positive definite. The sketching equations can be thought of as 'tangent' equations that require $B_{k+1}$ to incorporate information about the Hessian $\nabla^2 f(x_{k+1})$ at the most recent point $x_{k+1}$, as opposed to information about the average of $\nabla^2 f(x)$ between two points, i.e, along a secant. Consequently, in terms of the information used, the block updating formula is Newton-like rather than secant-like. A Hessian-vector product $\nabla^2 f(x_{k+1}) s_k^{(i)}$ can generally be computed much faster than the full Hessian $\nabla^2 f(x_{k+1})$, and the operation of computing $\nabla^2 f(x_{k+1}) s_k^{(i)}$ for multiple directions $s_k^{(1)}, \ldots, s_k^{(q)}$ can be done in parallel.

Computing the Hessian-vector products $\nabla^2 f(x_{k+1}) s_k^{(i)}$, referred to as *Hessian actions*, involves additional work beyond that of classical BFGS updates, where the gradients can be reused to compute $\nabla f(x_{k+1}) - \nabla f(x_k)$. However, the increased cost of block updates may be justified in order to obtain better search directions, for the same reason that Newton's method often outperforms gradient descent: the greater cost per iteration is compensated by convergence in fewer iterations, in regions where the curvature can be used effectively. Our numerical experiments in Section 7 explored this trade-off, and we found that using block updates did result in performance gains on many problems.

Other experiments indicate that quasi-Newton methods using Hessian actions and block updates are promising for empirical risk minimization problems arising from machine learning. Byrd, Hansen, Nocedal, and Yuan [32] proposed a stochastic limited-memory algorithm *Stochastic Quasi-Newton* (SQN), in which the secant equation is replaced by a sub-sampled sketching equation $B_{k+1} s_k = \widehat{\nabla}^2 f(x_{k+1}) s_k$ (here $\widehat{\nabla}^2 f(x)$ denotes a sub-sampled Hessian). The authors [32] remark that using the sub-sampled Hessian action avoids harmful effects from gradient differencing in the stochastic setting. In [31], a stochastic limited-memory method *Stochastic Block L-BFGS*, using block updates, outperformed other state-of-the-art methods when applied to large-scale logistic regression problems.

In this paper, we introduce a deterministic quasi-Newton method *Block BFGS*. The key fea-

ture of Block BFGS is the inclusion of information about $\nabla^2 f(x)$ along multiple directions, by enforcing that $B_{k+1}$ satisfies the sketching equations for a subset of previous search directions. We show that this method, performed with inexact Armijo-Wolfe line searches, has the same convergence properties as the classical BFGS method. Namely, if $f$ is twice differentiable, convex, and bounded below, and the gradient of $f$ is Lipschitz continuous, then Block BFGS converges. If, in addition, $f$ is strongly convex and the Hessian of $f$ is Lipschitz continuous, then Block BFGS achieves $Q$-superlinear convergence. Note that we use a slightly modified notion of $Q$-superlinear convergence: we prove that the sequence of quotients $\|x_k^{(i+1)} - x_*\|/\|x_k^{(i)} - x_*\|$, with possibly a small number of terms removed, converges to 0. The precise statement of this result is given in Theorem 2.5.1. We also note that our convergence results can easily be extended to block versions of the restricted Broyden class of quasi-Newton methods as in [27].

These results fill a gap in the theory of quasi-Newton methods, as updates based on the Hessian action have previously only been used within limited-memory methods, for which the analysis is significantly simpler. Because of its limited-memory nature, the Stochastic Block L-BFGS method in [31] is only proved to be $R$-linearly convergent (in expectation, when using a fixed step size). For this method, as is the case for the deterministic L-BFGS method [40], the convergence rate that is proved is worse than the rate for gradient descent (GD), even though in practice, L-BFGS almost always converges far more rapidly than GD. We believe that our proof of the $Q$-superlinear convergence of Block BFGS in this paper provides a rationale for the superior performance of the Stochastic Block L-BFGS method, and behavior of deterministic limited-memory Block BFGS methods as well.

Block BFGS can also be applied to non-convex functions. We show that if $f$ has bounded Hessian, then Block BFGS converges to a stationary point of $f$. Modified forms of the classical BFGS method also have natural extensions to block updates, so modified block quasi-Newton methods are applicable in the non-convex setting.

The paper is organized as follows. Section 2.2 contains preliminaries and describes Armijo-Wolfe inexact line searches. In Section 2.3, we formally define the Block BFGS method and several

variants. In Section 2.4 and Section 2.5 respectively, we show that Block BFGS converges, and converges superlinearly, for $f$ satisfying appropriate conditions. In Section 2.6, we show that Block BFGS converges for suitable non-convex functions, and describe several other modifications to adapt Block BFGS for non-convex optimization. In Section 2.7, we present the results of numerical experiments for several classes of convex and non-convex problems.

### 2.2 Preliminaries

The following notation will be used. The objective function of $n$ variables is denoted by $f : \mathbb{R}^n \to \mathbb{R}$. We write $g(x)$ for the gradient $\nabla f(x)$ and $G(x)$ for the Hessian $\nabla^2 f(x)$. For a sequence $\{x_k\}$, $f_k = f(x_k)$ and $g_k = g(x_k)$. However, we deliberately use $G_k = G(x_{k+1})$ to simplify the update formula.

The norm $\|\cdot\|$ denotes the $L_2$ norm, or for matrices, the $L_2$ operator norm. The Frobenius norm will be explicitly indicated as $\|\cdot\|_F$. Angle brackets $\langle \cdot, \cdot \rangle$ denote the standard inner product $\langle x, y \rangle = y^T x$ and the trace inner product $\langle X, Y \rangle = \mathrm{Tr}(Y^T X)$. We use either notation $\langle x, y \rangle$ or $y^T x$ as is convenient. The symbol $\Sigma^n$ denotes the space of $n \times n$ symmetric matrices, and $\preceq$ denotes the Löwner partial order; hence $A \succ 0$ means $A$ is positive definite.

An $L\Sigma L^T$ decomposition is a factorization of a positive definite matrix into a product $L\Sigma L^T$, where $L$ is lower triangular with ones on the diagonal, and $\Sigma = \mathrm{Diag}(\sigma_1^2, \ldots, \sigma_n^2)$. This is commonly called an $LDL^T$ decomposition in the literature, but we write $\Sigma$ in place of $D$ as we use $D$ to denote a matrix whose columns are previous search directions.

In the pseudocode for our algorithm, $\texttt{size}(A, 1)$ and $\texttt{size}(A, 2)$ denote the number of rows and columns of a matrix $A$ respectively. The $ij$-entry of a matrix $A$ will be denoted by $A_{ij}$. We use $\mathrm{Col}(A)$ to denote the linear space spanned by the columns of $A$. By convention, a sum over an empty index set is equal to 0.

Our inexact line search selects step sizes $\lambda_k$ satisfying the *Armijo-Wolfe* conditions: for param-

eters $\alpha, \beta$ with $0 < \alpha < \frac{1}{2}$ and $\alpha < \beta < 1$, the step satisfies

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \alpha \lambda_k \langle g_k, d_k \rangle \tag{2.2.1}$$

$$\langle g(x_k + \lambda_k d_k), d_k \rangle \geq \beta \langle g_k, d_k \rangle. \tag{2.2.2}$$

Furthermore, our line search always selects $\lambda_k = 1$ whenever this step size is admissible. This is important in the analysis of superlinear convergence in Section 2.5.

## 2.3 Block quasi-Newton Methods

In this section, we introduce *Block BFGS*, a quasi-Newton method with block updates, and several variants.

### 2.3.1 Block BFGS

**Algorithm 1** Block BFGS

**input**: $x_1^{(1)}, B_1, q$
**for** $k = 1, 2, 3 \ldots$ **do**
  **for** $i = 1, \ldots, q$ **do**
    $d_k^{(i)} \leftarrow -B_k^{-1} g_k^{(i)}$
    $\lambda_k^{(i)} \leftarrow \text{LINESEARCH}(x_k^{(i)}, d_k^{(i)})$
    $s_k^{(i)} \leftarrow \lambda_k^{(i)} d_k^{(i)}$
    $x_k^{(i+1)} \leftarrow x_k^{(i)} + s_k^{(i)}$
  **end for**
  $G_k \leftarrow G(x_k^{(q+1)})$
  $S_k \leftarrow [s_k^{(1)} \ldots s_k^{(q)}]$
  $D_k \leftarrow \text{FILTERSTEPS}(S_k, G_k)$
  **if** $D_k$ is not empty **then**
    $B_{k+1} \leftarrow B_k - B_k D_k (D_k^T B_k D_k)^{-1} D_k^T B_k + G_k D_k (D_k^T G_k D_k)^{-1} D_k^T G_k$
  **else**
    $B_{k+1} \leftarrow B_k$
  **end if**
  $x_{k+1}^{(1)} \leftarrow x_k^{(q+1)}$
**end for**

**Algorithm 2** FILTERSTEPS

**input**: $S_k, G_k$ **output**: $D_k$ **parameters**: threshold $\tau > 0$

Initialize $D_k \leftarrow S_k, i \leftarrow 1$
**while** $i \leq \texttt{size}(D_k, 2)$ **do**
  $\sigma_i^2 \leftarrow [D_k^T G_k D_k]_{ii} - \sum_{j=1}^{i-1} L_{ij}^2 \Sigma_{jj}$
  $s_i \leftarrow$ column $i$ of $D_k$
  **if** $\sigma_i^2 \geq \tau \|s_i\|^2$ **then**
    $\Sigma_{ii} \leftarrow \sigma_i^2$
    $L_{ii} \leftarrow 1$
    **for** $j = i+1, \ldots, \texttt{size}(D_k, 2)$ **do**
      $L_{ji} \leftarrow \frac{1}{\Sigma_{ii}}([D_k^T G_k D_k]_{ji} - \sum_{k=1}^{i-1} L_{ik} L_{jk} \Sigma_{kk})$
    **end for**
    $i \leftarrow i + 1$
  **else**
    Delete column $i$ from $D_k$ and row $i$ from $L$
  **end if**
**end while**

Block BFGS (Algorithm 1) takes $q$ steps in each block, using a fixed Hessian approximation $B_k$. We may also take a varying number of steps, bounded above by $q$, but we assume every block contains $q$ steps to simplify the presentation. We use a subscript $k$ for the block index, and superscripts $(i)$ for the steps within each block. The $k$-th block contains the iterates $x_k^{(1)}, \ldots, x_k^{(q+1)}$, and $x_{k+1}^{(1)} = x_k^{(q+1)}$. At each point $x_k^{(i)}$, the step direction is $d_k^{(i)} = -B_k^{-1} g_k^{(i)}$, and line search is performed to obtain a step size $\lambda_k^{(i)}$. We take a step $s_k^{(i)} = \lambda_k^{(i)} d_k^{(i)}$. The angle between $s_k^{(i)}$ and $-g_k^{(i)}$ is denoted $\theta_k^{(i)}$. As $B_k$ is positive definite, $\theta_k^{(i)} \in [0, \frac{\pi}{2})$.

After taking $q$ steps, the matrix $B_k$ is updated. Let $G_k = G(x_k^{(q+1)})$ denote the Hessian at the final iterate, and form the matrix $S_k = [s_k^{(1)} \ldots s_k^{(q)}]$. We apply the FILTERSTEPS procedure (Algorithm 2) to $S_k$, which returns a subset $D_k$ of the columns of $S_k$ satisfying $\sigma_i^2 \geq \tau \|s_i\|^2$, where $s_i$ is the $i$-th column of $D_k$ and $\sigma_i^2$ is the $i$-th diagonal entry of the $L\Sigma L^T$ decomposition of $D_k^T G_k D_k$. $\tau > 0$ is a parameter which controls the strictness of the filtering; a small value of $\tau$ permits $D_k$ to contain steps that are closer to being linearly dependent, as well as steps with smaller curvature. In essence, FILTERSTEPS iteratively computes the $L\Sigma L^T$ decomposition of $S_k^T G_k S_k$ and discards columns of $S_k$ corresponding to small diagonal entries, with the remaining columns forming $D_k$.

Define $q_k$ to be the number of columns of $D_k$. If $D_k$ is the empty matrix (all columns were removed), then no update is performed and $B_{k+1} = B_k$. If $D_k$ is not empty, the matrix $B_k$ is updated to have the same action as the Hessian $G_k$ on the column space of $D_k$, or equivalently,

$$B_{k+1}D_k = G_kD_k. \tag{2.3.1}$$

Let $D = D_k, G = G_k$. The formula for the update is given by

$$B_{k+1} = B_k - B_kD(D^TB_kD)^{-1}D^TB_k + GD(D^TGD)^{-1}D^TG. \tag{2.3.2}$$

This formula is invariant under a change of basis of $\mathrm{Col}(D_k)$, so we can choose $D_k$ to be any matrix with the same column space. To see this, observe that a change of basis is given by $D_kP$ for an invertible $q \times q$ matrix $P$. The update (2.3.2) for the matrix $D_kP$ is given by

$$\begin{aligned} B_{k+1} &= B_k - B_kDP(P^TD^TB_kDP)^{-1}P^TD^TB_k + GDP(P^TD^TGDP)^{-1}P^TD^TG \\ &= B_k - B_kD(D^TB_kD)^{-1}D^TB_k + GD(D^TGD)^{-1}D^TG. \end{aligned}$$

On the other hand, the matrix $D_k$ obtained from filtering $S_k$ is *not* invariant under a change of basis of $S_k$, and it is possible to control the number of columns removed by selecting an appropriate basis for $S_k$. We chose to take $S_k = [s_k^{(1)} \dots s_k^{(q)}]$ in order to retain control over the ratio $\det(D_k^TG_kD_k)/\det(D_k^TB_kD_k)$, which is crucial for our theoretical analysis. We also note that in [31], two other choices for the columns of $D_k$ were studied for use in the Stochastic Block L-BFGS method, and the results reported there showed that the choice $D_k = [s_k^{(1)} \dots s_k^{(q)}]$ worked best.

As is the case for standard quasi-Newton updates, there are many possible updates that satisfy equation (2.3.1). The specific Block BFGS update (2.3.2) is derived as follows. Let $H_k = B_k^{-1}$ be the approximation of the inverse Hessian. In contrast with the classical BFGS update, the update (2.3.2) is chosen so that $H_{k+1}$ is the nearest matrix to $H_k$ in a weighted norm, satisfying the system of sketching equations $H_{k+1}G_kD_k = D_k$ rather than a set of secant equations. That is, $H_{k+1}$ is the

solution to the minimization problem

$$\begin{aligned} \min_{\widetilde{H}\in\mathbb{R}^{n\times n}} \quad & \|\widetilde{H} - H_k\|_{G_k} \\ \text{s.t} \quad & \widetilde{H} = \widetilde{H}^T, \widetilde{H}G_kD_k = D_k \end{aligned} \tag{2.3.3}$$

where $\|\cdot\|_{G_k}$ is the norm $\|X\|_{G_k} = \mathrm{Tr}(XG_kX^TG_k)$. This norm is induced by an inner product, so $H_{k+1}$ is an orthogonal projection onto the subspace $\{\widetilde{H} \in \Sigma^n : \widetilde{H}G_kD_k = D_k\}$. In analogy with the classical BFGS update, $H_{k+1}$ has a simple formula in terms of block updates, which was obtained in [30].

**Theorem 2.3.1.** *The Block BFGS update of $H_k$ is given by*

$$H_{k+1} = D(D^TGD)^{-1}D^T + (I - D(D^TGD)^{-1}D^TG)H_k(I - GD(D^TGD)^{-1}D^T). \tag{2.3.4}$$

Taking the inverse yields formula (2.3.2). Moreover, as shown in [30], we have

**Lemma 2.3.2.** *If $B_k$ ($H_k$) and $D_k^TG_kD_k$ are positive definite, then the Block BFGS update (2.3.2) for $B_{k+1}$ ((2.3.4) for $H_{k+1}$) is positive definite.*

*Proof.* Our proof is adapted from Theorem 3.1 of [30]. Let $z \in \mathbb{R}^n$, and define $w = D_k^Tz, v = z - G_kD_k(D_k^TG_kD_k)^{-1}w$. Using formula (2.3.4), we find that

$$z^TH_{k+1}z = w^T(D_k^TG_kD_k)^{-1}w + v^TH_kv$$

so $z^TH_{k+1}z \geq 0$. Furthermore, $z^TH_{k+1}z = 0$ only if both $w = 0$ and $v = 0$, in which case $z = 0$. Hence $H_{k+1}$ is positive definite. □

In Section 2.4, we show that Block BFGS converges even if $B_k = B_{k+1} = \ldots$ is stationary. In Section 2.5, we show that when $f$ is strongly convex, the parameter $\tau$ can be naturally chosen so an update is always performed, and the convergence is superlinear.

In theory, FILTERSTEPS is required to ensure that the update (2.3.2) exists. However, in practice, one is unlikely to encounter linearly dependent directions, or directions lying exactly in the

null space of $G_k$. Thus, one may omit FILTERSTEPS unless there is reason to believe that $G_k$ is singular and problems will arise. However, filtering may improve numerical stability, by removing nearly linearly dependent steps from $D_k$.

### 2.3.2 Rolling Block BFGS

Block BFGS uses the same matrix $B_k$ throughout each block of $q$ steps. We could also add information from these steps immediately, at the cost of doing far more updates. This variant, *Rolling Block BFGS*, performs a block update after every step, using a subset $D_k$ of the previous $q$ steps. $D_k$ is formed by adding $s_k$ as the first column of $D_{k-1}$, removing $s_{k-q}$ if present, and filtering.

In general, one might consider schemes for interleaving standard BFGS updates with periodic block updates, to capture additional second-order information.

## 2.4 Convergence of Block BFGS

In this section we prove that Block BFGS with inexact Armijo-Wolfe line searches converges under the same conditions as does the classical BFGS method. These conditions are given in Assumption 1.

**Assumption 1**

1. $f$ is convex, twice differentiable, and bounded below.
2. For all $x$ in the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$, the Hessian satisfies $G(x) \preceq MI$, or equivalently, $g(x)$ is Lipschitz continuous with Lipschitz constant $M$.

The main goal of this section is to prove the following theorem. The concept of our proof is similar to the analysis given by Powell [41] for the classical BFGS method.

**Theorem 2.4.1.** *Let $f$ be a function satisfying Assumption 1, and let $\{x_k\}_{k=1}^{\infty}$ denote the sequence of all iterates produced by Block BFGS. Then* $\liminf_k \|g_k\| = 0$.

We begin by proving several lemmas. The first two are well known; see [27, 41].

**Lemma 2.4.2.** *$\sum_{k=1}^{\infty}\langle -g_k, s_k\rangle < \infty$, and therefore $\langle -g_k, s_k\rangle \to 0$.*

*Proof.* From the Armijo condition (2.2.1), $\langle -g_k, s_k\rangle = \lambda_k\langle -g_k, d_k\rangle \leq (1/\alpha)(f_k - f_{k+1})$. As $f$ is bounded below,

$$\sum_{k=1}^{\infty}\langle -g_k, s_k\rangle \leq (1/\alpha)\sum_{k=1}^{\infty}(f_k - f_{k+1}) \leq (1/\alpha)(f_1 - \lim_{k\to\infty} f_k) < \infty.$$

□

**Lemma 2.4.3.** *If the gradient $g(x)$ is Lipschitz continuous with constant $M$, then for $c_1 = \frac{1-\beta}{M}$, we have $\|s_k\| \geq c_1\|g_k\|\cos\theta_k$.*

*Proof.* Let $y_k = g_{k+1} - g_k$. From the Wolfe condition (2.2.2),

$$\langle y_k, s_k\rangle = \langle g_{k+1}, s_k\rangle - \langle g_k, s_k\rangle \geq (1-\beta)\langle -g_k, s_k\rangle.$$

By the Lipschitz continuity of the gradient, $\|y_k\| \leq M\|s_k\|$. Therefore

$$(1-\beta)\|g_k\|\|s_k\|\cos\theta_k = (1-\beta)\langle -g_k, s_k\rangle \leq \langle y_k, s_k\rangle \leq M\|s_k\|^2$$

yielding $\|s_k\| \geq c_1\|g_k\|\cos\theta_k$. □

It is possible that $D_k$ is empty for all $k \geq k_0$, and no further updates are made to $B_{k_0}$. This may occur, for example, if $G(x)$ has arbitrarily small eigenvalues and $\tau$ is chosen to be large. In this case, Block BFGS is equivalent to a scaled gradient method $x_{k+1} = x_k - \lambda_k B_{k_0}^{-1} g_k$ with $B_{k_0}$ a constant positive-definite matrix, for all $k \geq k_0$, which is well-known to converge to a stationary point.

For the remainder of this section, we assume that there is an infinite sequence of updates. In fact, we may further assume that an update is made for every $k$, as one can verify that the propositions of this section continue to hold when we restrict our arguments to the subsequence

of $\{B_k\}$ for which updates are made. This simplifies the notation. Note, however, that the same cannot simply be assumed in Section 2.5. The results in that section do *not* hold if updates are skipped. However, in Section 2.5 we are able to choose $\tau$ so as to guarantee that an update is made for every $k$.

**Lemma 2.4.4.** *Let $c_3 = \mathrm{Tr}(B_1) + qM$. Then for all $k$,*

$$\mathrm{Tr}(B_k) \leq c_3 k \quad \textit{and} \quad \sum_{j=1}^{k} \mathrm{Tr}(D_j^T B_j^2 D_j (D_j^T B_j D_j)^{-1}) \leq c_3 k$$

*Proof.* Clearly $\mathrm{Tr}(B_1) \leq c_3$. Define $E_j = G_j^{\frac{1}{2}} D_j$, and let $P_j = E_j(E_j^T E_j)^{-1} E_j^T$ be the orthogonal projection onto $\mathrm{Col}(E_j)$, so that $G_j D_j (D_j^T G_j D_j)^{-1} D_j^T G_j = G_j^{\frac{1}{2}} P_j G_j^{\frac{1}{2}}$. For $k \geq 1$, we expand $\mathrm{Tr}(B_{k+1})$ using Equation (2.3.2):

$$\begin{aligned} 0 < \mathrm{Tr}(B_{k+1}) &= \mathrm{Tr}(B_1) + \sum_{j=1}^{k} \mathrm{Tr}(G_j^{\frac{1}{2}} P_j G_j^{\frac{1}{2}}) - \sum_{j=1}^{k} \mathrm{Tr}(D_j^T B_j^2 D_j (D_j^T B_j D_j)^{-1}) \\ &\leq \mathrm{Tr}(B_1) + k(qM) - \sum_{j=1}^{k} \mathrm{Tr}(D_j^T B_j^2 D_j (D_j^T B_j D_j)^{-1}) \end{aligned}$$

where the first inequality follows from the positive definiteness of $B_{k+1}$ (Lemma 2.3.2) and the second inequality follows since $\mathrm{rank}(P_j) \leq q$, and $\|G_j^{\frac{1}{2}} P_j G_j^{\frac{1}{2}}\| \leq \|G_j\| \|P_j\| \leq M$. This shows $\mathrm{Tr}(B_{k+1}) \leq c_3(k+1)$ and $\sum_{j=1}^{k} \mathrm{Tr}(D_j^T B_j^2 D_j (D_j^T B_j D_j)^{-1}) \leq c_3 k$. □

**Lemma 2.4.5.** *Let $s_k^{(i)}$ be a step included in $D_k$. Then*

$$\frac{\lambda_k^{(i)} \|g_k^{(i)}\|^2}{\langle -g_k^{(i)}, s_k^{(i)} \rangle} \leq \mathrm{Tr}(D_k^T B_k^2 D_k (D_k^T B_k D_k)^{-1})$$

*Proof.* By the Gram-Schmidt process applied to the columns of $D_k$, we can find a set of $B_k$-conjugate vectors $\{v_1, \ldots, v_{q_k}\}$ spanning $\mathrm{Col}(D_k)$ with $v_1 = s_k^{(i)}$. Using the matrix $[v_1 \ldots v_{q_k}]$ for

$D_k$, we have

$$D_k^T B_k D_k = \mathrm{Diag}(\langle s_k^{(i)}, -\lambda_k^{(i)} g_k^{(i)} \rangle, \langle v_2, B_k v_2 \rangle, \ldots, \langle v_{q_k}, B_k v_{q_k} \rangle)$$

and therefore

$$\begin{aligned}\mathrm{Tr}(D_k^T B_k^2 D_k (D_k^T B_k D_k)^{-1}) &= \sum_{\ell=1}^{q_k} [D_k^T B_k^2 D_k]_{\ell\ell} [D_k^T B_k D_k]_{\ell\ell}^{-1} \\ &= \frac{(\lambda_k^{(i)} \|g_k^{(i)}\|)^2}{\lambda_k^{(i)} \langle -g_k^{(i)}, s_k^{(i)} \rangle} + \sum_{\ell=2}^{q_k} \frac{\|B_k v_\ell\|^2}{\langle v_\ell, B_k v_\ell \rangle} \geq \frac{\lambda_k^{(i)} \|g_k^{(i)}\|^2}{\langle -g_k^{(i)}, s_k^{(i)} \rangle}\end{aligned}$$

□

We may assume without loss of generality that $D_k = [s_k^{(1)} \ldots s_k^{(q_k)}]$.

**Corollary 2.4.6.**

$$\prod_{j=1}^{k} \prod_{i=1}^{q_j} \frac{\lambda_j^{(i)} \|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)} \rangle} \leq (qc_3)^{qk}$$

*Proof.* Let $\widehat{q}_k = \sum_{j=1}^{k} q_j$, and note that $k \leq \widehat{q}_k \leq qk$. Hence, from Lemmas 2.4.4 and 2.4.5,

$$\frac{1}{\widehat{q}_k} \sum_{j=1}^{k} \sum_{i=1}^{q_j} \frac{\lambda_j^{(i)} \|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)} \rangle} \leq \frac{qk}{\widehat{q}_k} c_3 \leq q c_3$$

Applying the arithmetic mean-geometric mean (AM-GM) inequality,

$$\left( \prod_{j=1}^{k} \prod_{i=1}^{q_j} \frac{\lambda_j^{(i)} \|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)} \rangle} \right) \leq (qc_3)^{\widehat{q}_k} \leq (qc_3)^{qk}.$$

□

**Lemma 2.4.7.** $\det(B_k) \leq \left(\frac{c_3 k}{n}\right)^n$ *for all* $k$.

*Proof.* By Lemma 2.4.4, $\mathrm{Tr}(B_k) \leq c_3 k$. Recall that the trace is equal to the sum of the eigenvalues, and the determinant to the product. Applying the AM-GM inequality to the eigenvalues of $B_k$, we obtain $\det(B_k) \leq \left(\frac{c_3 k}{n}\right)^n$. □

We will need the following two classical results from matrix theory; see [42].

**Sylvester's Determinant Identity** Let $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{m\times n}$. Then

$$\det(I_n + AB) = \det(I_m + BA)$$

**Sherman-Morrison-Woodbury Formula** Let $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{k\times k}$ be invertible, and $U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{k\times n}$. If $A + UCV$ and $C^{-1} + VA^{-1}U$ are invertible, then $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$.

**Lemma 2.4.8.**

$$\det(B_{k+1}) = \frac{\det(D_k^T G_k D_k)}{\det(D_k^T B_k D_k)} \det(B_k)$$

*Proof.* Let $B = B_k, B^+ = B_{k+1}, D = D_k, G = G_k$. Then

$$\det(B^+) = \det(B)\det(I + B^{-\frac{1}{2}}GD(D^TGD)^{-1}D^TGB^{-\frac{1}{2}} - B^{\frac{1}{2}}D(D^TBD)^{-1}D^TB^{\frac{1}{2}}).$$

Define $X = B^{-\frac{1}{2}}GD(D^TGD)^{-1}D^TGB^{-\frac{1}{2}}$ and $Y = D^TGD + D^TGB^{-1}GD$. Note that $I + X$ is invertible since $X \succeq 0$ and $I \succ 0$, and $Y$ is invertible since $D^TGD \succ 0$. Thus, we can write

$$\det(B^+) = \det(B)\det(I + X)\det(I - (I + X)^{-1}B^{\frac{1}{2}}D(D^TBD)^{-1}D^TB^{\frac{1}{2}}).$$

Applying Sylvester's determinant identity to each term,

$$\det(I + X) = \det(I + (D^TGB^{-\frac{1}{2}})(B^{-\frac{1}{2}}GD(D^TGD)^{-1})) = \det(Y)\det(D^TGD)^{-1}$$
$$\det(I - (I + X)^{-1}B^{\frac{1}{2}}D(D^TBD)^{-1}D^TB^{\frac{1}{2}}) = \det(I - D^TB^{\frac{1}{2}}(I + X)^{-1}B^{\frac{1}{2}}D(D^TBD)^{-1})$$

Applying the Sherman-Morrison-Woodbury formula to $I+X$ with $U = B^{-\frac{1}{2}}GD, C = (D^TGD)^{-1}, V = D^TGB^{-\frac{1}{2}}$, we obtain $(I + X)^{-1} = I - B^{-\frac{1}{2}}GDY^{-1}D^TGB^{-\frac{1}{2}}$, so

$$\det(I - (I + X)^{-1}B^{\frac{1}{2}}D(D^TBD)^{-1}D^TB^{\frac{1}{2}}) = \det(D^TGD)^2\det(Y)^{-1}\det(D^TBD)^{-1}.$$

Thus $\det(B^+) = \det(B)\det(D^TGD)\det(D^TBD)^{-1}$ as desired. □

**Lemma 2.4.9.**

$$\det(B_{k+1}) \geq \left(\prod_{i=1}^{q_k} \frac{1}{\lambda_i}\right)(\tau c_1)^{q_k}\det(B_k)$$

*Proof.* Recall that the columns of $D_k$ satisfy $\sigma_i^2 \geq \tau\|s_k^{(i)}\|^2$, where $\sigma_i$ is the $i$-th diagonal element of the $L\Sigma L^T$ decomposition of $D_k^TG_kD_k$. We have $\det(D_k^TG_kD_k) = \prod_{i=1}^{q_k}\sigma_i^2$ and $\det(D_k^TB_kD_k) \leq \prod_{i=1}^{q_k}[D_k^TB_kD_k]_{ii} = \prod_{i=1}^{q_k}\langle s_k^{(i)}, -\lambda_k^{(i)}g_k^{(i)}\rangle$. By Lemma 2.4.8,

$$\begin{aligned}\det(B_{k+1}) &= \det(B_k)\frac{\det(D_k^TG_kD_k)}{\det(D_k^TB_kD_k)} \\ &\geq \det(B_k)\frac{\prod_{i=1}^{q_k}\tau\|s_k^{(i)}\|^2}{\prod_{i=1}^{q_k}\langle s_k^{(i)}, -\lambda_k^{(i)}g_k^{(i)}\rangle} \geq \det(B_k)\prod_{i=1}^{q_k}\frac{\tau}{\lambda_k^{(i)}}\frac{\|s_k^{(i)}\|}{\|g_k^{(i)}\|\cos\theta_k^{(i)}}.\end{aligned}$$

By Lemma 2.4.3, $\frac{\|s_k^{(i)}\|}{\|g_k^{(i)}\|\cos\theta_k^{(i)}} \geq c_1$. Hence $\det(B_{k+1}) \geq \left(\prod_{i=1}^{q_k}\frac{1}{\lambda_k^{(i)}}\right)(\tau c_1)^{q_k}\det(B_k)$. □

**Corollary 2.4.10.**

$$\det(B_{k+1}) \geq (\tau c_1)^{qk}\det(B_1)\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{1}{\lambda_j^{(i)}}$$

**Corollary 2.4.11.** *There exists a constant $c_4$ such that for all $k$,*

$$\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle} \leq c_4^k$$

*Proof.* Multiplying the inequalities of Corollary 2.4.6 and Lemma 2.4.7, we obtain

$$\left(\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\lambda_j^{(i)}\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle}\right)\left(\frac{\det(B_{k+1})}{\det(B_1)}\right) \leq (qc_3)^{qk}\left(\frac{(c_3(k+1)/n)^n}{\det(B_1)}\right) \leq \rho_1^k$$

for some constant $\rho_1$. Using the lower bound of Corollary 2.4.10, we also obtain

$$\left(\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\lambda_j^{(i)}\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle}\right)\left(\frac{\det(B_{k+1})}{\det(B_1)}\right) \geq \left(\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\lambda_j^{(i)}\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle}\right)\cdot(\tau c_1)^{qk}\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{1}{\lambda_j^{(i)}}$$
$$= (\tau c_1)^{qk}\left(\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle}\right)$$

Take $c_4 = \frac{\rho_1}{(\tau c_1)^q}$, whence $\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle} \leq c_4^k$. □

Finally, we can establish our main result.

*Proof.* (of Theorem 2.4.1) Assume to the contrary that $\|g_k^{(i)}\|$ is bounded away from zero. Lemma 2.4.2 implies that $\langle g_k^{(i)}, -s_k^{(i)}\rangle \to 0$. Thus, there exists $k_0$ such that for $k \geq k_0$, $\frac{\|g_k^{(i)}\|^2}{\langle g_k^{(i)}, -s_k^{(i)}\rangle} > c_4 + 1$. This contradicts Corollary 2.4.11, as $\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\|g_j^{(i)}\|^2}{\langle -g_j^{(i)}, s_j^{(i)}\rangle} \leq c_4^k$ for all $k$. We conclude that $\liminf_k \|g_k\| = 0$. □

A similar analysis shows that Rolling Block BFGS (Section 2.3.2) converges.

**Theorem 2.4.12.** *Assume $f$ satisfies Assumption 1. Then the sequence $\{g_k\}_{k=1}^{\infty}$ produced by Rolling Block BFGS satisfies* $\liminf_k \|g_k\| = 0$.

*Proof.* By the calculations for Corollary 2.4.6, we have $\prod_{j=1}^{k}\frac{\lambda_j\|g_j\|^2}{\langle -g_j, s_j\rangle} \leq c_3^k$.

$D_k$ is produced by adding column $s_k$ to $D_{k-1}$, removing $s_{k-q}$ if present, and then running Algorithm 2. Without loss of generality, assume that $D_k = [s_k \ldots s_{k-q_k+1}]$. By definition, $B_k$ satisfies $B_k D_{k-1} = G_{k-1}D_{k-1}$. Thus, we have

$$\det(D_k^T B_k D_k) \leq \prod_{i=0}^{q_k-1}\langle s_{k-i}, B_k s_{k-i}\rangle = \langle s_k, B_k s_k\rangle \prod_{i=1}^{q_k-1}\langle s_{k-i}, G_{k-1}s_{k-i}\rangle$$

which gives an analogue of Lemma 2.4.9:

$$\det(B_{k+1}) \geq \frac{\prod_{i=0}^{q_k-1}\tau\|s_{k-i}\|^2}{\langle s_k, -\lambda_k g_k\rangle\prod_{i=1}^{q_k-1}\langle s_{k-i}, G_{k-1}s_{k-i}\rangle}\det(B_k) \geq \frac{1}{\lambda_k}\frac{c_1\tau^q}{M^{q-1}}\det(B_k).$$

Thus $\det(B_{k+1}) \geq \left(\frac{c_1 \tau^q}{M^{q-1}}\right)^k \det(B_1) \prod_{j=1}^{k} \frac{1}{\lambda_k}$. The remainder of the proof follows exactly as in the proofs of Corollary 2.4.11 and Theorem 2.4.1. $\square$

## 2.5 Superlinear Convergence of Block BFGS

In this section we show that Block BFGS converges $Q$-superlinearly under the same conditions as does BFGS, namely, that $f$ is strongly convex in a neighborhood of its minimizer, and its Hessian is Lipschitz continuous. We use the characterization of superlinear convergence given by Dennis and Moré [43], and employ an argument similar to the analysis used by Griewank and Toint [44] for partitioned quasi-Newton updates. The following conditions, which strengthen Assumption 1, will apply to $f$ throughout this section.

**Assumption 2**

1. $f$ is convex and twice differentiable, with $G(x) \preceq MI$ on the level set $\{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$.

2. $f$ has a minimizer $x_*$ for which $G(x_*)$ is non-singular. Note that this implies $x_*$ is unique.

3. $G(x)$ is Lipschitz in a neighborhood of $x_*$, with Lipschitz constant $\mu$.

Since Assumption 2 is stronger than Assumption 1, Theorem 2.4.1 implies that the iterates produced by Block BFGS converge to the unique stationary point $x_*$. The continuity of $G(x)$ and the fact that $G(x_*)$ is non-singular imply that $f$ is strongly convex in a neighborhood $S$ of $x_*$. Superlinear convergence is an asymptotic property, so we may restrict our attention to the tail of the sequence $\{x_k\}$, contained in $S$. That is, we may assume without loss of generality that all iterates $\{x_k\}$ lie in a region $S$ on which $f$ is strongly convex, with

$$mI \preceq G(x) \preceq MI \qquad \forall x \in S$$

for constants $0 < m \leq M$.

In this section, we assume $\tau \leq m$, where $\tau$ is the parameter in FILTERSTEPS. Since $\sigma_1^2 = [S_k^T G_k S_k]_{11} = \langle s_k^{(1)}, G_k s_k^{(1)} \rangle \geq m\|s_k^{(1)}\|^2$, the first column of $D_k$ is never removed by FILTERSTEPS. This guarantees that an update is always performed. The choice of $\tau$ is important and can impact superlinear convergence; we give a detailed discussion in the remarks concluding this section.

**Theorem 2.5.1.** *Let $f$ be a function satisfying Assumption 2. Block BFGS converges $Q$-superlinearly along the subsequence of steps in $D_k$; that is,*

$$\lim_{\substack{k\to\infty \\ i\in D_k}} \frac{\|x_k^{(i+1)} - x_*\|}{\|x_k^{(i)} - x_*\|} = 0.$$

To clarify the statement of this theorem, the quotients $\|x_k^{(i+1)} - x_*\|/\|x_k^{(i)} - x_*\|$ in the subsequence are those for which $s_k^{(i)}$ is in $D_k$. If every step is included in $D_k$, then we have $Q$-superlinear convergence for the sequence of points $\{x_k^{(i)}\}$ in the usual sense. To give an example of the contrary, suppose the step $s_{10}^{(2)}$ is removed by filtering; then the quotient $\|x_{10}^{(3)} - x_*\|/\|x_{10}^{(2)} - x_*\|$ is not captured in the subsequence. In theory, one step is guaranteed per block $D_k$, but we note that in practice, $D_k$ contains all or nearly all steps for every $k$.

We begin by showing that Block BFGS converges $R$-linearly. The first three lemmas are well known; see [27, 41]. These three lemmas apply to every step, and thus we write $x_{k+1}$ for the iterate immediately following $x_k$, instead of using superscripts.

**Lemma 2.5.2.** *For $c_1 = \frac{1-\beta}{M}$ and $c_2 = \frac{2(1-\alpha)}{m}$,*

$$c_1\|g_k\|\cos\theta_k \leq \|s_k\| \leq c_2\|g_k\|\cos\theta_k$$

*Proof.* By Taylor's theorem, there exists a point $\widetilde{x}$ on the line segment joining $x_k, x_{k+1}$ such that $f(x_{k+1}) = f(x_k) + \langle g_k, s_k\rangle + \frac{1}{2}s_k^T G(\widetilde{x})s_k$. From (2.2.1), $f(x_{k+1}) - f(x_k) \leq \alpha\langle g_k, s_k\rangle$, so $(1-\alpha)\langle -g_k, s_k\rangle \geq \frac{1}{2}s_k^T G(\widetilde{x})s_k \geq \frac{1}{2}m\|s_k\|^2$. Rearranging yields $\|s_k\| \leq c_2\|g_k\|\cos\theta_k$. The lower bound was shown in Lemma 2.4.3. □

This next lemma is known as the *Polya-Łojasiewicz inequality* and is standard [41]. We give an alternate proof here.

**Lemma 2.5.3.** *For any* $x \in S$, $\|g(x)\|^2 \geq 2m(f(x) - f_*)$.

*Proof.* The result is immediate if $x = x_*$, so assume $x \neq x_*$. By Taylor's theorem, there exists a point $\widetilde{x}$ on the line segment joining $x, x_*$ such that $f(x_*) = f(x) + g(x)^T(x_* - x) + \frac{1}{2}(x_* - x)^T G(\widetilde{x})(x_* - x)$, in which case

$$g(x)^T(x - x_*) = f(x) - f_* + \frac{1}{2}(x_* - x)^T G(\widetilde{x})(x_* - x) \geq f(x) - f_* + \frac{1}{2}m\|x - x_*\|^2.$$

Using the Cauchy-Schwarz inequality, we find that $\|g(x)\|\|x - x_*\| \geq f(x) - f_* + \frac{1}{2}m\|x - x_*\|^2$. Applying the AM-GM inequality and squaring yields $\|g(x)\|^2 \geq 2m(f(x) - f_*)$. □

**Lemma 2.5.4.**

$$f_{k+1} - f_* \leq (1 - 2\alpha m c_1 \cos^2\theta_k)(f_k - f_*)$$

*Proof.* The Armijo condition (2.2.1) and Lemma 2.5.2 imply that

$$f_{k+1} - f_k \leq \alpha\langle g_k, s_k\rangle = -\alpha\|g_k\|\|s_k\|\cos\theta_k \leq -\alpha c_1\|g_k\|^2\cos^2\theta_k.$$

By Lemma 2.5.3, $\|g_k\|^2 \geq 2m(f_k - f_*)$. Hence $f_{k+1} - f_* \leq (1 - 2\alpha m c_1 \cos^2\theta_k)\,(f_k - f_*)$. □

Define $r_k = \|x_k^{(q+1)} - x_*\|$. $R$-linear convergence implies that the errors $r_k$ diminish to zero rapidly enough that $\sum_{k=1}^{\infty} r_k < \infty$, a key property.

**Theorem 2.5.5.** *There exists* $\delta < 1$ *such that* $f(x_k^{(q+1)}) - f_* \leq \delta^k(f(x_1^{(1)}) - f_*)$, *and thus* $\sum_{k=1}^{\infty} r_k < \infty$.

*Proof.* From Lemma 2.4.11, $\prod_{j=1}^{k}\prod_{i=1}^{q_j}\frac{\|g_j^{(i)}\|}{\|s_j^{(i)}\|\cos\theta_j^{(i)}} \leq c_4^k$. Lemma 2.5.2 gives the upper bound $\|s_j^{(i)}\| \leq c_2\|g_j^{(i)}\|\cos\theta_j^{(i)}$. Substituting, we find

$$\prod_{j=1}^{k}\prod_{i=1}^{q_j}\cos^2\theta_j^{(i)} \geq \left(\frac{1}{c_2^q c_4}\right)^k.$$

From this, we see that at least $\frac{1}{2}k$ of the angles must satisfy $\cos^2\theta_j^{(i)} \geq \left(\frac{1}{c_2^q c_4}\right)^2$.

By Lemma 2.5.4, $f(x_k^{(i+1)}) - f_* \leq (1 - 2\alpha m c_1 \cos^2\theta_k)(f(x_k^{(i)}) - f_*)$. Using our bound on the angles,

$$f(x_k^{(q+1)}) - f_* \leq \left(1 - 2\alpha m c_1 \left(\frac{1}{c_2^q c_4}\right)^2\right)^{\frac{1}{2}k} (f(x_1^{(1)}) - f_*).$$

Hence, we may take $\delta = \left(1 - \frac{2\alpha m c_1}{c_2^{2q} c_4^2}\right)^{1/2}$. The strong convexity of $f$ implies that $\frac{1}{2}m\|x - x_*\|^2 \leq f(x) - f_* \leq \frac{1}{2}M\|x - x_*\|^2$, so we have $r_k \leq (\sqrt{\delta})^k \sqrt{\frac{M}{m}}\|x_1^{(1)} - x_*\|$. Therefore $\sum_{k=1}^{\infty} r_k < \infty$. □

The classical BFGS method is invariant under a linear change of coordinates. It is easy to verify that Block BFGS also has this invariance, so we may assume without loss of generality that $G(x_*) = I$. This greatly simplifies the following calculations. Given that Theorem 2.4.1 implies that Block BFGS converges, we will also assume that the iterates lie in the region around $x_*$ where $G(x)$ is Lipschitz continuous.

**Lemma 2.5.6.** *For any $v \in \mathbb{R}^n$, $\|(G_k - I)v\| \leq \mu r_k \|v\|$.*

*Proof.* Since $G(x_*) = I$,

$$\|(G_k - I)v\| \leq \|G(x_k^{(q+1)}) - G(x_*)\|\|v\| \leq \mu\|x_k^{(q+1)} - x_*\|\|v\| = \mu r_k\|v\|.$$

□

The following notion is useful in our analysis. Define $\widetilde{B}_{k+1}$ to be the matrix obtained by performing a Block BFGS update on $B_k$ with $G_k = G(x_*)$. Since we assumed $G(x_*) = I$, we have the explicit formula

$$\widetilde{B}_{k+1} = B_k - B_k D_k (D_k^T B_k D_k)^{-1} D_k^T B_k + D_k (D_k^T D_k)^{-1} D_k^T$$

and its inverse $\widetilde{H}_{k+1}$ is given by

$$\widetilde{H}_{k+1} = D_k (D_k^T D_k)^{-1} D_k^T + (I - D_k (D_k^T D_k)^{-1} D_k^T) H_k (I - D_k (D_k^T D_k)^{-1} D_k^T).$$

**Lemma 2.5.7.** *Let $B = B_k, \widetilde{B} = \widetilde{B}_{k+1}, D = D_k$. Define the following orthogonal projections:*

1. $P = B^{\frac{1}{2}} D(D^T BD)^{-1} D^T B^{\frac{1}{2}}$, *the projection onto* $\mathrm{Col}(B^{\frac{1}{2}} D)$.
2. $P_D = D(D^T D)^{-1} D^T$, *the projection onto* $\mathrm{Col}(D)$.
3. $P_B = BD(D^T B^2 D)^{-1} D^T B$, *the projection onto* $\mathrm{Col}(BD)$.

*Then*

$$\|B - I\|_F^2 - \|\widetilde{B} - I\|_F^2 = \|P_B - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 + 2\,\mathrm{Tr}(B(B^{\frac{1}{2}} P B^{\frac{1}{2}}) - (B^{\frac{1}{2}} P B^{\frac{1}{2}})^2)$$

*Furthermore,* $\mathrm{Tr}(B(B^{\frac{1}{2}} P B^{\frac{1}{2}}) - (B^{\frac{1}{2}} P B^{\frac{1}{2}})^2) \geq 0$, *and thus* $\|\widetilde{B} - I\|_F \leq \|B - I\|_F$.

*Proof.* Expand the Frobenius norm and use the identity $\mathrm{Tr}(BP_D) = \mathrm{Tr}(B^{\frac{1}{2}} P B^{\frac{1}{2}} P_D)$ to obtain

$$\begin{aligned}
\|B - I\|_F^2 - \|\widetilde{B} - I\|_F^2 &= 2\,\mathrm{Tr}(B(B^{\frac{1}{2}} P B^{\frac{1}{2}})) - \mathrm{Tr}((B^{\frac{1}{2}} P B^{\frac{1}{2}})^2) - 2\,\mathrm{Tr}(B^{\frac{1}{2}} P B^{\frac{1}{2}}) \\
&\quad - \mathrm{Tr}(P_D^2) + 2\,\mathrm{Tr}(P_D) \\
&= 2\,\mathrm{Tr}(B(B^{\frac{1}{2}} P B^{\frac{1}{2}})) - 2\,\mathrm{Tr}((B^{\frac{1}{2}} P B^{\frac{1}{2}})^2) \\
&\quad + \mathrm{Tr}((B^{\frac{1}{2}} P B^{\frac{1}{2}})^2) - 2\,\mathrm{Tr}(B^{\frac{1}{2}} P B^{\frac{1}{2}}) + \mathrm{Tr}(I) \\
&\quad - \mathrm{Tr}(P_D^2) + 2\,\mathrm{Tr}(P_D) - \mathrm{Tr}(I)
\end{aligned}$$

Factoring the above equation produces

$$\|B - I\|_F^2 - \|\widetilde{B} - I\|_F^2 = \|I - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 - \|I - P_D\|_F^2 + 2\,\mathrm{Tr}(B(B^{\frac{1}{2}} P B^{\frac{1}{2}}) - (B^{\frac{1}{2}} P B^{\frac{1}{2}})^2).$$

Let $P_B^{\perp}$ be the projection onto the orthogonal complement of $\mathrm{Col}(BD)$; hence $I = P_B + P_B^{\perp}$. Since $\langle P_B^{\perp}, B^{\frac{1}{2}} P B^{\frac{1}{2}} \rangle = \mathrm{Tr}(P_B^{\perp} BD(D^T BD)^{-1} D^T B) = 0$, we have $\|I - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 = \|P_B - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 + \|P_B^{\perp}\|_F^2$. The Frobenius norm of an orthogonal projection is equal to the square root of its rank, and thus

$$\|I - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 - \|I - P_D\|_F^2 = \|P_B - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 + \|P_B^{\perp}\|_F^2 - \|I - P_D\|_F^2 = \|P_B - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2$$

This gives the desired equation. Now, observe that

$$\begin{aligned}\mathrm{Tr}(B(B^{\frac{1}{2}}PB^{\frac{1}{2}}) - (B^{\frac{1}{2}}PB^{\frac{1}{2}})^2) &= \mathrm{Tr}(BPB(I-P)) \\ &= \mathrm{Tr}((I-P)BPB(I-P)) \geq 0\end{aligned}$$

where in the second equality we have used that $I-P$ is the orthogonal projection onto $\mathrm{Col}(B^{\frac{1}{2}}D)^{\perp}$, and is therefore idempotent. This proves $\|\widetilde{B} - I\|_F \leq \|B - I\|_F$. □

Intuitively, $\widetilde{B}_{k+1}$ and $\widetilde{H}_{k+1}$ should be closer approximations of $I$ than $B_k$ and $H_k$. This is made precise in the next lemma.

**Lemma 2.5.8.** $\|\widetilde{B}_{k+1} - I\|_F \leq \|B_k - I\|_F$ *and* $\|\widetilde{H}_{k+1} - I\|_F \leq \|H_k - I\|_F$.

*Proof.* That $\|\widetilde{B}_{k+1} - I\|_F \leq \|B_k - I\|_F$ was shown in Lemma 2.5.7. Clearly $\|\widetilde{H}_{k+1} - I\|_F \leq \|H_k - I\|_F$, as $\widetilde{H}_{k+1}$ is defined as the orthogonal projection of $H_k$ onto the subspace of matrices $\{\widetilde{H} \in \Sigma^n : \widetilde{H}D_k = D_k\}$, which contains $I$ (see (2.3.3)). □

**Lemma 2.5.9.** *There exists an index $k_0$ and constants $\kappa_1, \kappa_2$ such that $\|B_{k+1} - \widetilde{B}_{k+1}\|_F \leq \kappa_1 r_k$ and $\|H_{k+1} - \widetilde{H}_{k+1}\|_F \leq (\|H_k - I\|_F + 1)\kappa_2 r_k$ for all $k \geq k_0$.*

*Proof.* Define $\Delta_k = (G_k - I)D_k$. For brevity, let $\widetilde{B} = \widetilde{B}_{k+1}, \widetilde{H} = \widetilde{H}_{k+1}, H = H_k, D = D_k, G = G_k$, and $\Delta = \Delta_k$. We may assume the columns of $D$ are orthonormal, so $D^TD = I$. By Lemma 2.5.6, every column $\delta_i$ of $\Delta$ satisfies $\|\delta_i\| \leq \mu r_k$, which gives the useful bounds $\|\Delta\|, \|\Delta^T\| \leq \mu\sqrt{q}r_k$. This stems from the fact that a matrix $A$ of rank $q$ satisfies $\|A\| = \|A^T\| \leq \|A\|_F \leq \sqrt{q}\|A\|$, which we will use frequently.

To prove the first inequality, we write

$$\begin{aligned}\|B_{k+1} - \widetilde{B}\|_F &= \|GD(D^TGD)^{-1}D^TG - DD^T\|_F \\ &= \|GD(I + D^T\Delta)^{-1}D^TG - DD^T\|_F.\end{aligned}$$

By the Sherman-Morrison-Woodbury formula, $(I + D^T\Delta)^{-1} = I - D^T(I + \Delta D^T)^{-1}\Delta$. Let $X = I + \Delta D^T$. Inserting this expression and using the triangle inequality, we have

$$\begin{aligned}\|GD(I + D^T\Delta)^{-1}D^TG - DD^T\|_F &= \|GDD^TG - DD^T - GDD^TX^{-1}\Delta D^TG\|_F \\ &\leq \|GDD^TG - DD^T\|_F + \|GDD^TX^{-1}\Delta D^TG\|_F\end{aligned}$$

By a routine calculation,

$$\|GDD^TG - DD^T\|_F = \|\Delta\Delta^T + \Delta D^T + D\Delta^T\|_F,$$

hence $\|GDD^TG - DD^T\|_F \leq \rho_2 r_k$ for some constant $\rho_2$.

To bound the Frobenius norm of the other term, we bound its operator norm. Since $\Delta_k \to 0$ as $r_k \to 0$, there exists an index $k_0$ such that for $k \geq k_0$,

1. $\|X - I\| \leq \frac{1}{2}$, so $\|X^{-1}\| \leq 2$, and
2. $\|G - I\| \leq 1$, so $\|G\| \leq 2$

in which case $\|GDD^TX^{-1}\Delta D^TG\| \leq \rho_3 r_k$ for some $\rho_3$. Taking $\kappa_1 = \rho_2 + \sqrt{q}\rho_3$, we then have $\|B_{k+1} - \widetilde{B}\|_F \leq \kappa_1 r_k$ for all $k \geq k_0$.

A similar analysis applies to $\|H_{k+1} - \widetilde{H}\|_F$. Using the triangle inequality,

$$\begin{aligned}\|H_{k+1} - \widetilde{H}\|_F \leq\ & \|D(D^TGD)^{-1}D^T - DD^T\|_F \\ &+ \|(D(D^TGD)^{-1}D^TG - DD^T)H + H(GD(D^TGD)^{-1}D^T - DD^T)\|_F \\ &+ \|D(D^TGD)^{-1}D^TGHGD(D^TGD)^{-1}D^T - DD^THDD^T\|_F\end{aligned}$$

We bound each of the three terms. As before, $(D^TGD)^{-1} = I - D^TX^{-1}\Delta$, so we have $\|D(D^TGD)^{-1}D^T - DD^T\|_F = \|DD^TX^{-1}\Delta D^T\|_F$. For $k \geq k_0$, $\|X^{-1}\| \leq 2$, so $\|D(D^TGD)^{-1}D^T - DD^T\|_F \leq \rho_4 r_k$ for some $\rho_4$.

For the second term, observe that

$$GD(D^TGD)^{-1}D^T - DD^T = \Delta D^T - DD^TX^{-1}\Delta D^T - \Delta DX^{-1}\Delta D^T.$$

Hence, the norm of the second term is bounded above by $\rho_5 r_k \|H\|$ for some $\rho_5$.

Finally, we bound the operator norm of the third term. Factoring out $D$ and $D^T$ on the left and right, we can write the inside term as

$$D^TGHGD - D^THD - (D^TX^{-1}\Delta D^TGHGD + D^TGHGDD^TX^{-1}\Delta) + D^TX^{-1}\Delta D^TGHGDD^TX^{-1}\Delta.$$

Since $D^TGHGD - D^THD = \Delta^THD + D^TH\Delta + \Delta^TH\Delta$, the operator norm of the third term is bounded above by $\rho_6 r_k \|H\|$ for some $\rho_6$. Adding together the three terms, there is a constant $\kappa_2$ with $\|H_{k+1} - \widetilde{H}\|_F \leq (\|H_k - I\|_F + 1)\kappa_2 r_k$. □

Since superlinear convergence is an asymptotic property, we may assume $k_0 = 1$ in Lemma 2.5.9. We will also need the following technical result from [43].

**Lemma 2.5.10** (3.3 of [43])**.** *Let $\{\nu_k\}$ and $\{\delta_k\}$ be sequences of non-negative numbers such that $\nu_{k+1} \leq (1+\delta_k)\nu_k + \delta_k$ and $\sum_{k=1}^{\infty} \delta_k < \infty$. Then $\{\nu_k\}$ converges.*

**Corollary 2.5.11.** *$\{\|B_k - I\|_F\}_{k=1}^{\infty}$ and $\{\|H_k - I\|_F\}_{k=1}^{\infty}$ converge, and are therefore uniformly bounded. As an immediate corollary, $\{\|B_k\|_F\}_{k=1}^{\infty}$ and $\{\|H_k\|_F\}_{k=1}^{\infty}$ are also uniformly bounded.*

*Proof.* By Lemma 2.5.8 and Lemma 2.5.9, we have

$$\|H_{k+1} - I\|_F \leq \|H_{k+1} - \widetilde{H}_{k+1}\|_F + \|\widetilde{H}_{k+1} - I\|_F \leq (1 + \kappa_2 r_k)\|H_k - I\|_F + \kappa_2 r_k$$

Set $\nu_k = \|H_k - I\|_F$ and $\delta_k = \kappa_2 r_k$ in Lemma 2.5.10. Since $\sum_{k=1}^{\infty} r_k < \infty$, the sequence $\{\|H_k - I\|_F\}$ converges. The same reasoning applies to $\{\|B_k - I\|_F\}$. □

**Lemma 2.5.12.** *Recall the notation introduced in Lemma 2.5.7: $P_k$ is the orthogonal projection onto $\mathrm{Col}(B_k^{\frac{1}{2}} D_k)$, and $P_{B_k}$ the orthogonal projection onto $\mathrm{Col}(B_k D_k)$. Define the quantities $\varphi_k, \psi_k$ to be*

$$\varphi_k = \|P_{B_k} - B_k^{\frac{1}{2}} P_k B_k^{\frac{1}{2}}\|_F^2$$
$$\psi_k = \mathrm{Tr}(B_k(B_k^{\frac{1}{2}} P_k B_k^{\frac{1}{2}}) - (B_k^{\frac{1}{2}} P_k B_k^{\frac{1}{2}})^2)$$

*Then $\lim_{k\to\infty} \varphi_k = 0$ and $\lim_{k\to\infty} \psi_k = 0$.*

*Proof.* We first bound $\|\widetilde{B}_{k+1} - I\|_F^2$ in terms of $\|B_{k+1} - I\|_F^2$. By Lemma 2.5.9, $\|B_{k+1} - \widetilde{B}_{k+1}\|_F \leq \kappa_1 r_k$. Let $\kappa_3 = 2\kappa_1 \max_k\{\|B_k - I\|_F\}$; by Corollary 2.5.11, the maximum exists. Using the triangle inequality, we have

$$\begin{aligned}
\|\widetilde{B}_{k+1} - I\|_F^2 &\geq (\|B_{k+1} - I\|_F - \|B_{k+1} - \widetilde{B}_{k+1}\|_F)^2 \\
&= \|B_{k+1} - I\|_F^2 - 2\|B_{k+1} - I\|_F \|B_{k+1} - \widetilde{B}_{k+1}\|_F + \|B_{k+1} - \widetilde{B}_{k+1}\|_F^2 \\
&\geq \|B_{k+1} - I\|_F^2 - \kappa_3 r_k.
\end{aligned}$$

By Lemma 2.5.7, $\|B_k - I\|_F^2 - \|\widetilde{B}_{k+1} - I\|_F^2 \geq 0$. Summing over $k$ and telescoping, we find that

$$\begin{aligned}
\sum_{k=1}^{\infty} \left(\|B_k - I\|_F^2 - \|\widetilde{B}_{k+1} - I\|_F^2\right) &\leq \sum_{k=1}^{\infty} \left(\|B_k - I\|_F^2 - \|B_{k+1} - I\|_F^2\right) + \kappa_3 r_{k+1} \\
&\leq \|B_1 - I\|_F^2 + \kappa_3 \sum_{k=1}^{\infty} r_{k+1} < \infty
\end{aligned}$$

from which we deduce that $\|B_k - I\|_F^2 - \|\widetilde{B}_{k+1} - I\|_F^2 \to 0$. Expressed in terms of $\varphi_k$ and $\psi_k$, Lemma 2.5.7 states that $\|B_k - I\|_F^2 - \|\widetilde{B}_{k+1} - I\|_F^2 = \varphi_k + 2\psi_k$ and $\varphi_k, \psi_k \geq 0$. Hence $\varphi_k, \psi_k$ converge to 0. □

Note that Lemma 2.5.12 does not imply that $\|B_k - I\|_F \to 0$, since it is possible for $\limsup \|\widetilde{B}_{k+1} - I\|_F > 0$. It is well-known that for the classical BFGS method, the Hessian approximation $B_k$ might not converge to the Hessian at the optimal solution.

**Lemma 2.5.13.** *For any $w_k \in \mathrm{Col}(D_k)$,*

$$\left(1 - \frac{w_k^T B_k^2 w_k}{w_k^T B_k w_k}\right)^2 \leq \varphi_k \quad \textit{and} \quad 0 \leq \frac{w_k^T B_k^3 w_k}{w_k^T B_k w_k} - \left(\frac{w_k^T B_k^2 w_k}{w_k^T B_k w_k}\right)^2 \leq \varphi_k + \psi_k,$$

*where $\varphi_k, \psi_k$ are defined in Lemma 2.5.12. Consequently, for any sequence $\{w_k\}_{k=1}^{\infty}$ with $w_k \in \mathrm{Col}(D_k)$, we have $\lim_{k\to\infty} \frac{w_k^T B_k^2 w_k}{w_k^T B_k w_k} = 1$ and $\lim_{k\to\infty} \frac{w_k^T B_k^3 w_k}{w_k^T B_k w_k} = 1$.*

*Proof.* For a fixed $k$, let $B = B_k, D = D_k$, and let $\Delta = (D^T B^2 D)^{-1} - (D^T B D)^{-1}$. Recall the definitions of $P, P_B$ from Lemma 2.5.7. We can write

$$\begin{aligned}\varphi_k = \|P_B - B^{\frac{1}{2}} P B^{\frac{1}{2}}\|_F^2 = \mathrm{Tr}((BD\Delta D^T B)^2) &= \mathrm{Tr}(D^T B^2 D \Delta D^T B^2 D \Delta) \\ &= \mathrm{Tr}((I - D^T B^2 D (D^T B D)^{-1})^2)\end{aligned}$$

Take a $B_k$-orthogonal basis $\{v_1, \ldots, v_{q_k}\}$ for $\mathrm{Col}(D_k)$ with $v_1 = w_k$. The $i$-th diagonal entry of $(I - D^T B^2 D (D^T B D)^{-1})^2$ is then

$$\left(1 - \frac{v_i^T B^2 v_i}{v_i^T B v_i}\right)^2 + \sum_{j \neq i} \frac{(v_i^T B^2 v_j)^2}{v_i^T B v_i v_j^T B v_j}$$

Since every term is non-negative, we conclude that $\left(1 - \frac{w_k^T B^2 w_k}{w_k^T B w_k}\right)^2 \leq \varphi_k$, which proves the first statement. Also, notice that $\sum_{i=1}^{q_k} \sum_{j \neq i} \frac{(v_i^T B^2 v_j)^2}{v_i^T B v_i v_j^T B v_j} \leq \varphi_k$.

Next, write $\mathrm{Tr}(B(B^{\frac{1}{2}} P B^{\frac{1}{2}})) = \mathrm{Tr}(D^T B^3 D (D^T B D)^{-1})$ and $\mathrm{Tr}((B^{\frac{1}{2}} P B^{\frac{1}{2}})^2) = \mathrm{Tr}((D^T B^2 D (D^T B D)^{-1})^2)$. Again taking a $B_k-$orthogonal basis $\{v_1, \ldots, v_{q_k}\}$, we have

$$\begin{aligned}\mathrm{Tr}(D^T B^3 D (D^T B D)^{-1}) &= \sum_{i=1}^{q_k} \frac{v_i^T B^3 v_i}{v_i^T B v_i} \\ \mathrm{Tr}((D^T B^2 D (D^T B D)^{-1})^2) &= \sum_{i=1}^{q_k} \left(\frac{v_i^T B^2 v_i}{v_i^T B v_i}\right)^2 + \sum_{i=1}^{q_k} \sum_{j \neq i} \frac{(v_i^T B^2 v_j)^2}{v_i^T B v_i v_j^T B v_j}\end{aligned}$$

Thus

$$\begin{aligned}\operatorname{Tr}(B(B^{\frac{1}{2}}PB^{\frac{1}{2}}) - (B^{\frac{1}{2}}PB^{\frac{1}{2}})^2) &= \sum_{i=1}^{q_k}\left(\frac{v_i^TB^3v_i}{v_i^TBv_i} - \left(\frac{v_i^TB^2v_i}{v_i^TBv_i}\right)^2\right) - \sum_{i=1}^{q_k}\sum_{j\neq i}\frac{(v_i^TB^2v_j)^2}{v_i^TBv_iv_j^TBv_j} \\ &\geq \sum_{i=1}^{q_k}\left(\frac{v_i^TB^3v_i}{v_i^TBv_i} - \left(\frac{v_i^TB^2v_i}{v_i^TBv_i}\right)^2\right) - \varphi_k\end{aligned}$$

By the Cauchy-Schwarz inequality applied to $v^TB^2v = \langle B^{\frac{1}{2}}v, B^{\frac{3}{2}}v\rangle$, we have $\frac{v^TB^3v}{v^TBv} \geq \left(\frac{v^TB^2v}{v^TBv}\right)^2$ for every $v \in \mathbb{R}^n$. Hence $0 \leq \frac{w_k^TB^3w_k}{w_k^TBw_k} - \left(\frac{w_k^TB^2w_k}{w_k^TBw_k}\right)^2 \leq \varphi_k + \psi_k$. The limits then follow from Lemma 2.5.12, since $\varphi_k, \psi_k \to 0$. □

**Corollary 2.5.14.** *Given any* $w_k \in \operatorname{Col}(D_k)$*,*

$$\frac{\|(B_k - I)w_k\|}{\|w_k\|} \leq \sqrt{2\varphi_k + \psi_k}$$

*Consequently, for any sequence* $\{w_k\}_{k=1}^{\infty}$ *with* $w_k \in \operatorname{Col}(D_k)$*,*

$$\lim_{k\to\infty}\frac{\|(B_k - I)w_k\|}{\|w_k\|} = 0$$

*Proof.* By Lemma 2.5.13 and a routine calculation,

$$\begin{aligned}\frac{\|B_k^{\frac{1}{2}}(B_k - I)w_k\|}{\|B_k^{\frac{1}{2}}w_k\|} &= \sqrt{\frac{w_k^TB_k^3w_k}{w_k^TB_kw_k} - 2\frac{w_k^TB_k^2w_k}{w_k^TB_kw_k} + 1} \\ &= \sqrt{\frac{w_k^TB_k^3w_k}{w_k^TB_kw_k} - \left(\frac{w_k^TB_k^2w_k}{w_k^TB_kw_k}\right)^2 + \left(1 - \frac{w_k^TB_k^2w_k}{w_k^TB_kw_k}\right)^2} \\ &\leq \sqrt{2\varphi_k + \psi_k}\end{aligned}$$

Since $\{\|B_k\|\}, \{\|H_k\|\}$ are uniformly bounded by Corollary 2.5.11, the result follows. □

**Lemma 2.5.15.** *A step size of* $\lambda_k = 1$ *is eventually admissible for steps* $d_k$ *included in* $D_k$*.*

*Proof.* We check that $\lambda_k = 1$ satisfies the Armijo-Wolfe conditions for all sufficiently large $k$. Let $\alpha$ and $\beta$ be the Armijo-Wolfe parameters and choose a constant $\gamma$ such that $0 < \gamma < \frac{\frac{1}{2}-\alpha}{1-\alpha}$. By

Corollary 2.5.14, for all sufficiently large $k$, the steps $d_k \in \mathrm{Col}(D_k)$ satisfy

$$\frac{\|(B_k - I)d_k\|}{\|d_k\|} \leq \gamma \tag{2.5.1}$$

in which case $\langle g_k, d_k\rangle = \langle g_k + d_k, d_k\rangle - \|d_k\|^2 \leq -(1-\gamma)\|d_k\|^2$.

By Taylor's theorem, there exists a point $\widetilde{x}_k$ on the line segment joining $x_k, x_k + d_k$ with $f(x_k + d_k) = f(x_k) + \langle g_k, d_k\rangle + \frac{1}{2} d_k^T G(\widetilde{x}_k) d_k$. Since $f(x_k) \leq f(x_{k-1}^{(q+1)})$, the strong convexity of $f$ implies that $\|x_k - x_*\| \leq \sqrt{M/m}\, r_{k-1}$. Hence, taking $\rho_7 = \mu\sqrt{M/m}$, we have $\|G(\widetilde{x}_k) - I\| \leq \mu\|\widetilde{x}_k - x_*\| \leq \rho_7(r_{k-1} + \|d_k\|)$. For the step size $\lambda_k = 1$,

$$\begin{aligned} f(x_k + d_k) - f(x_k) &= \alpha\langle g_k, d_k\rangle + (1-\alpha)\langle g_k, d_k\rangle + \frac{1}{2} d_k^T G(\widetilde{x}) d_k \\ &\leq \alpha\langle g_k, d_k\rangle - ((1-\alpha)(1-\gamma) - 1/2 - (\rho_7/2)(r_{k-1} + \|d_k\|))\, \|d_k\|^2 \end{aligned}$$

Since $(1-\alpha)(1-\gamma) - 1/2 > 0$ and $r_{k-1} + \|d_k\| \to 0$, a step size of $\lambda_k = 1$ satisfies the Armijo condition (2.2.1) for all sufficiently large $k$.

Next, apply Taylor's theorem to the function $t \mapsto \langle g(x_k + t d_k), d_k\rangle$ to obtain a point $\widetilde{x}_k$ on the line segment joining $x_k, x_k + d_k$ with $\langle g(x_k + d_k), d_k\rangle = \langle g_k, d_k\rangle + d_k^T G(\widetilde{x}_k) d_k$. Choosing $\gamma = \frac{\beta}{2-\beta}$ in (2.5.1), Corollary 2.5.14 implies that for sufficiently large $k$, $\langle -g_k, d_k\rangle = \langle g_k + d_k, -d_k\rangle + \|d_k\|^2 \leq (1 - \frac{1}{2}\beta)^{-1}\|d_k\|^2$. We can also take $k$ large enough so that $1 - \rho_7(r_{k-1} + \|d_k\|) \geq 0$, and we then have

$$\begin{aligned} \langle g(x_k + d_k), d_k\rangle &\geq \langle g_k, d_k\rangle + (1 - \rho_7(r_{k-1} + \|d_k\|))\|d_k\|^2 \\ &\geq (\beta/2 + (1 - \beta/2)\rho_7(r_{k-1} + \|d_k\|))\langle g_k, d_k\rangle \end{aligned}$$

Thus, the Wolfe condition (2.2.2) is satisfied for all sufficiently large $k$. □

Lemma 2.5.15 applies only to steps $d_k$ included in $D_k$. However, since Block BFGS does not prefer any particular step for inclusion in $D_k$, it is likely that eventually $\lambda_k = 1$ is admissible for *all* steps. This issue reveals a subtle artifact of the proof method, and we return to discuss it in the

remark after the following proof of Theorem 2.5.1.

*Proof.* (of Theorem 2.5.1) Let $s_k^{(i)}$ be any step included in $D_k$. To simplify the notation, we write $x = x_k^{(i)}, x^+ = x_k^{(i+1)}, g = g_k^{(i)}, g^+ = g_k^{(i+1)}$, and $d = d_k^{(i)}, s = s_k^{(i)}$. By Lemma 2.5.15, eventually $\lambda = 1$ is admissible for all steps in $D_k$, so $s = d$. From the triangle inequality, $\|d\| \leq \|x - x_*\| + \|x^+ - x_*\|$, so

$$\frac{\|g^+\|}{\|d\|} \geq \frac{m\|x^+ - x_*\|}{\|x - x_*\| + \|x^+ - x_*\|}. \tag{2.5.2}$$

Next, write

$$\begin{aligned}\frac{\|(B_k - I)d\|}{\|d\|} &= \frac{\|g(x+d) - g(x) - G(x_*)d - g(x+d)\|}{\|d\|} \\ &\geq \frac{\|g(x+d)\|}{\|d\|} - \frac{\|g(x+d) - g(x) - G(x_*)d\|}{\|d\|}.\end{aligned}$$

By continuity of the Hessian, the second term converges to 0. Thus, Corollary 2.5.14 implies that $\frac{\|g^+\|}{\|d\|} = \frac{\|g(x+d)\|}{\|d\|} \to 0$. We deduce from (2.5.2) that

$$\frac{\|x^+ - x_*\|}{\|x - x_*\|} \to 0.$$

Hence, we have $Q$-superlinear convergence along the subsequence of steps in $D_k$. □

The same argument, with minimal alteration, applies to Rolling Block BFGS.

Remarks

1. As we observed earlier, the choice to include $s_k^{(1)}$ in $D_k$ is arbitrary. The proof of Theorem 2.5.1 holds with *any* selection rule for $D_k$ as long as it guarantees $\sum_{k=1}^{\infty} r_k < \infty$. Therefore, it is likely that Theorem 2.5.1 and Lemma 2.5.15 apply to *all* steps. That is, eventually $\lambda_k = 1$ is admissible for all steps and $\frac{\|x_k^{(i+1)} - x_*\|}{\|x_k^{(i)} - x_*\|} \to 0$. In fact, by selecting $D_k$ in a particular way, we can ensure that eventually $\lambda_k = 1$ is admissible for all steps.

**Corollary 2.5.16.** *Suppose that $D_k$ is constructed to always contain a step for which $\lambda_k = 1$*

*is not admissible, whenever such a step exists in the $k$-th block. Then $\lambda_k = 1$ is eventually admissible for* all *steps.*

*Proof.* When executing the $k$-th update, we specifically set the first column of $D_k$ to a step $d_k$ from the $k$-th block for which $\lambda_k = 1$ is not admissible, if any such step exists. If we could find such a step $d_k$ for infinitely many $k$, then this process would produce an infinite sequence of steps $d_k \in \mathrm{Col}(D_k)$ for which $\lambda_k = 1$ is never eventually admissible. This contradicts Lemma 2.5.15. □

However, Corollary 2.5.16 does *not* show that in general, $\lambda_k = 1$ is eventually admissible for all steps, as it only holds when we select steps in an adversarial manner. This example highlights an interesting dichotomy arising from our proof method. On one hand, Theorem 2.5.1 and Lemma 2.5.15 are retrospective and apply to any sequence $\{D_k\}$ that we select. This strongly suggests that they should hold for all steps. On the other hand, the method of proof (based on analyzing the convergence of $\|B_k - I\|_F^2 - \|\widetilde{B}_{k+1} - I\|_F^2$) makes use only of the steps in $D_k$, and thus can only prove things about the steps in $D_k$.

2. The parameter $\tau$ has no equivalent in the classical BFGS method, and enforces a lower bound on the curvature of steps used in the update. If $\tau$ is chosen to be too large, then it is possible that $B_k$ is not updated on some iterations; in this case, the convergence rate will not be superlinear. A *sufficient* condition for $B_k$ to be updated on every iteration, and hence for superlinear convergence, is to take $\tau \leq m$, but this requires knowledge of a lower bound on $m$, the least eigenvalue of the Hessian.

   This issue can be avoided if $f$ is strongly convex on the entire level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$, by using a slightly modified version of FILTERSTEPS. Instead of $\tau$, the user selects any $\tau' > 0$. The first step $s_k^{(1)}$ is unconditionally included in $D_k$, and then subsequent steps $s_k^{(2)}, \ldots, s_k^{(q+1)}$ are included only if the condition $\sigma_i^2 > \tau' \|s_k^{(i)}\|^2$ holds. Since $\sigma_1^2 = \langle s_k^{(1)}, G_k s_k^{(1)} \rangle \geq m \|s_k^{(1)}\|^2$, every entry of $\Sigma$ satisfies $\sigma_i \geq \tau \|s_k^{(i)}\|^2$ for $\tau =$

$\min\{\tau', m\} > 0$, and thus the condition for convergence is satisfied. This guarantees $Q$-superlinear convergence for any choice of $\tau'$, although larger $\tau'$ reduces the number of steps in $D_k$ (see Theorem 2.5.1).

## 2.6 Modified Block BFGS for Non-Convex Optimization

Convergence theory for the classical BFGS method does not extend to non-convex functions. However, with minor modifications, BFGS performs well for non-convex optimization and can be shown to converge in some cases. Modifications that have been studied include:

1. Cautious Updates (Li and Fukushima, [45])

   A BFGS update is performed only if $\frac{y_k^T s_k}{\|s_k\|^2} \geq \epsilon \|g_k\|^\alpha$, where $\epsilon, \alpha$ are parameters.

2. Modified Updates (Li and Fukushima, [46])

   The secant equation is modified to $B_{k+1} s_k = z_k$, where $z_k = y_k + r_k s_k$ and the parameter $r_k$ is chosen so that $z_k^T s_k \geq \epsilon \|s_k\|^2$.

3. Damped BFGS (ell, [47])

   The secant equation is modified to $B_{k+1} s_k = z_k$, where $z_k = \theta_k y_k + (1 - \theta_k) B_k s_k$, and for $0 < \phi < 1$, the damping constant $\theta_k$ is determined by

$$\theta_k = \begin{cases} 1, & \text{if } y_k^T s_k \geq \phi s_k^T B_k s_k \\ \frac{(1-\phi) s_k^T B_k s_k}{s_k^T B_k s_k - y_k^T s_k}, & \text{otherwise} \end{cases}$$

   This is perhaps the most widely used modified BFGS method. Unfortunately, no global convergence proof is known for this method.

We show Block BFGS converges for non-convex functions, and describe analogous modifications for block updates. The next theorem provides a framework for proving convergence in the non-convex setting.

**Theorem 2.6.1.** *Assume $f$ is twice differentiable and $-MI \preceq G(x) \preceq MI$ for all $x$ in the convex hull of the level set $\{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$. Suppose that $\{\widetilde{G}_k\}_{k=1}^{\infty}$ is a sequence of symmetric matrices satisfying, for all $k$, the conditions*

1. $-MI \preceq \widetilde{G}_k \preceq MI$
2. *For some constant $\eta > 0$, the matrix $D_k$ produced by* FILTERSTEPS$(S_k, \widetilde{G}_k)$ *satisfies $D_k^T \widetilde{G}_k D_k \succeq \eta D_k^T D_k$*

*Then we may perform Block BFGS using the updates*

$$B_{k+1} = B_k - B_k D_k (D_k^T B_k D_k)^{-1} D_k^T B_k + \widetilde{G}_k D_k (D_k^T \widetilde{G}_k D_k)^{-1} D_k^T \widetilde{G}_k$$

*and Block BFGS converges in the sense that* $\liminf_k \|g_k\| = 0$.

*Proof.* The proof follows that of Theorem 2.4.1, with several changes. First, note that Lemma 2.3.2 implies that $B_{k+1}$ remains positive definite, since FILTERSTEPS ensures that $D_k^T \widetilde{G}_k D_k$ is positive definite. Observe that Lemma 2.4.3 continues to hold, as the condition $-MI \preceq G(x) \preceq MI$ for all $x$ in the convex hull of the level set implies that the gradient $g$ is Lipschitz with constant $M$. In Lemma 2.4.4, take the constant $c_3$ to be $c_3 = \operatorname{Tr}(B_1) + \frac{qM^2}{\eta}$ and notice that

$$\operatorname{Tr}(\widetilde{G}_j D_j (D_j^T \widetilde{G}_j D_j)^{-1} D_j^T \widetilde{G}_j) \leq \frac{1}{\eta} \operatorname{Tr}(\widetilde{G}_j D_j (D_j^T D_j)^{-1} D_j^T \widetilde{G}_j) \leq \frac{qM^2}{\eta}$$

where the last inequality follows because $D_j (D_j^T D_j)^{-1} D_j^T$ is the orthogonal projection onto $\operatorname{Col}(D_j)$ and has rank $q_j \leq q$, and $\|\widetilde{G}_j D_j (D_j^T D_j)^{-1} D_j^T \widetilde{G}_j\| \leq \|\widetilde{G}_j\|^2 = M^2$.

The remainder of the proof is similar to Theorem 2.4.1. □

**Lemma 2.6.2.** *Assume $f$ is twice differentiable and $-MI \preceq G(x) \preceq MI$ for all $x$ in the level set $\{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$. If $D_k^T G_k D_k$ satisfies $\sigma_i^2 \geq \tau \|s_i\|^2$, where $\sigma_i$ is the $i$-th diagonal entry of the $L\Sigma L^T$ decomposition of $D_k^T G_k D_k$, then $D_k^T G_k D_k \succeq \eta D_k^T D_k$ for $\eta = \frac{\tau^q}{q^q M^{q-1}}$.*

*Proof.* Let $G = G_k, D = D_k$. Without loss of generality, we may assume the columns of $D$ have norm 1, as otherwise we can normalize $D$ by right-multiplying by a positive diagonal matrix. Then the diagonal entries $\sigma_i^2$ of the $L\Sigma L^T$ decomposition of $D^T G D$ satisfy $\sigma_i^2 \geq \tau$.

Order the eigenvalues of $D^T G D$ as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_q > 0$. We have

$$\lambda_q = \frac{\det(D^T G D)}{\prod_{i=1}^{q-1} \lambda_i} \geq \frac{\tau^q}{(qM)^{q-1}}.$$

Since every column of $D$ has norm 1, the eigenvalues of $D^T D$ are bounded by $\mathrm{Tr}(D^T D) = q$. Hence $I \succeq \frac{1}{q} D^T D$ and so $D^T G D \succeq \frac{\tau^q}{(qM)^{q-1}} I \succeq \frac{\tau^q}{q^q M^{q-1}} D^T D$. □

Block BFGS (Algorithm 1) satisfies the conditions of Lemma 2.6.2 when we take $\widetilde{G}_k = G_k$ and apply FILTERSTEPS (Algorithm 2). Thus Theorem 2.6.1 shows that Block BFGS converges globally for non-convex functions. The filtering procedure is analogous to the cautious update (1) of Li and Fukushima, and hence, it is possible, although very unlikely, that filtering will produce an empty $D_k$. Hessian modification and Powell's damping method can also be extended to block updates.

### 2.7 Numerical Experiments

We evaluate the performance of several block quasi-Newton methods by generating a *performance profile* [48], which can be described as follows. Given a set of algorithms $\mathcal{S}$ and a set of problems $\mathcal{P}$, let $t_{s,p}$ be the cost for algorithm $s$ to solve problem $p$. For each problem $p$, let $m_p$ be the minimum cost to solve $p$ of any algorithm. A performance profile is a plot comparing the functions

$$\rho_s(r) = \frac{|\{p \in \mathcal{P} : t_{s,p}/m_p \leq r\}|}{|\mathcal{P}|}$$

for all $s \in \mathcal{S}$. Observe that $\rho_s(r)$ is the fraction of problems in $\mathcal{P}$ that algorithm $s$ solved within a factor $r$ of the cost of the best algorithm for problem $p$. As reference points, we include the classical BFGS method and gradient descent in $\mathcal{S}$.

For our inexact line search, we used the function `WolfeLineSearch` from *minFunc* [49], a mature and widely used Matlab library for unconstrained optimization. The line search parameters were $\alpha = 0.1$ and $\beta = 0.75$, and `WolfeLineSearch` was configured to use interpolation with an initial step size $\lambda = 1$ (options $\texttt{LS\_type} = 1, \texttt{LS\_init} = 0, \texttt{LS\_interp} = 1, \texttt{LS\_multi} = 0$).

From preliminary experiments, we found that large values of $q$ tend to increase numerical errors, eventually leading to search directions $d_k$ that are not descent directions. This effect is particularly pronounced when $q \geq \sqrt{n}$. The experiments in [31] also obtained the best performance when $\lfloor n^{1/4} \rfloor \leq q \leq \sqrt{n}$. In creating performance profiles, we opted for $q = \lfloor n^{1/3} \rfloor$.

### 2.7.1 Convex Experiments

We compared the methods listed below.

1. *BFGS*

2. *Block BFGS Variant 1*, or *B-BFGS1*

   Block BFGS (Algorithm 1). We store the full inverse Hessian approximation $H_k$ and compute $d_k = -H_k g_k$ by a matrix-vector product. We do not perform FILTERSTEPS, so the update (2.3.4) uses all steps.

3. *Block BFGS Variant 2*, or *B-BFGS2*

   Block BFGS (Algorithm 1), with Algorithm 2 and $\tau = 10^{-3}$. As in B-BFGS1, the full Hessian approximation $H_k$ is stored. $H_k$ is updated by (2.3.4) using the steps returned by Algorithm 2.

4. *Block BFGS with* $q = 1$, or *B-BFGS-q1*

   This compares the effect of using a single sketching equation as in Block BFGS updates versus using the standard secant equation of BFGS updates.

5. *Rolling Block BFGS*, or *RB-BFGS*

See Section 2.3.2. We take a smaller value $q = \min\{3, \lfloor n^{1/3} \rfloor\}$ for this method, and omit filtering.

6. *Gradient Descent*, or *GD*

Each algorithm is considered to have *solved* a problem when it reduces the objective value to less than some threshold $f_{stop}$. The thresholds $f_{stop}$ are pre-computed for each problem $p$ by minimizing $p$ with minFunc to obtain a near-optimal solution $f_*$, and setting $f_{stop} = f_* + 0.01|f_*|$.

We measure the cost $t_{s,p}$ in two metrics: the number of steps, and the amount of CPU time. Every step $s_k^{(i)}$ is counted once when measuring the number of steps.

**Logistic Regression Tests**

As in [31], we ran tests on *logistic regression* problems, a common classification technique in statistics. For our purposes, it suffices to describe the objective function. Given a set of $m$ data points $(y_i, x_i)$, where $y_i \in \{0, 1\}$ is the class, and $x_i \in \mathbb{R}^n$ is the vector of features of the $i$-th data point, we minimize, over all weights $w \in \mathbb{R}^n$, the loss function

$$L(w) = -\frac{1}{m}\sum_{i=1}^{m} \log \phi(y_i, x_i, w) + \frac{1}{2m} w^T Q w \tag{2.7.1}$$

$$\phi(y_i, x_i, w) = \begin{cases} \frac{1}{1+\exp(-x_i^T w)} & \text{if } y_i = 1 \\ 1 - \frac{1}{1+\exp(-x_i^T w)} & \text{if } y_i = 0 \end{cases}$$

where $Q \succ 0$ in the 'regularization' term. Figure 2.1 shows the performance profiles for this test. See Appendix 2.9 for a list of the data sets and our choices for $Q$.

In Figure 2.1, we see that the block methods B-BFGS1, B-BFGS2, and RB-BFGS all outperform BFGS in terms of the number of steps to completion. Considering the amount of CPU time used, B-BFGS1 is competitive with BFGS, while B-BFGS2 and RB-BFGS are more expensive than BFGS. This suggests that the additional curvature information added in block updates allows Block BFGS to find better search directions, but at the cost of the update operation being more

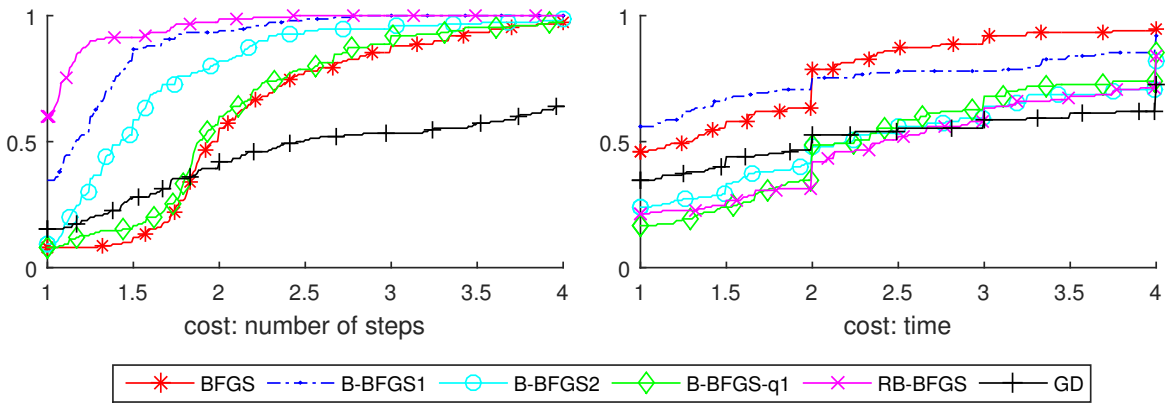


Figure 2.1: Logistic Regression profiles ($\rho_s(r)$)

expensive. B-BFGS-q1 and BFGS exhibit very similar performance when measured in steps, so there appears to be little difference between using a single sketching equation and a secant equation on this class of problems.

Interestingly, B-BFGS1 outperformed B-BFGS2, indicating that steps are being removed from the update, which would improve the search directions. The most likely explanation is that $\tau = 10^{-3}$ is excessively large relative to the eigenvalues of $G(x)$.

### Log Barrier QP Tests

We tested problems of the form

$$\min_{y \in \mathbb{R}^s} \; F(y) = \frac{1}{2} y^T \overline{Q} y + \overline{c}^T y - 1000 \sum_{i=1}^{n} \log(\overline{b} - \overline{A} y)_i \tag{2.7.2}$$

where $\overline{Q} \succeq 0$, $\overline{c} \in \mathbb{R}^s$, $\overline{b} \in \mathbb{R}^n$, and $\overline{A} \in \mathbb{R}^{n \times s}$. Note that the objective value is $+\infty$ if $y$ does not satisfy $\overline{A} y < \overline{b}$. In Appendix 2.9, we explain how to derive a log barrier problem from a QP in standard form. See Figure 2.2 for the performance profile. Note that problems with a barrier structure are atypical in the context of unconstrained minimization, and are usually solved with specific interior point methods. However, they are somewhat interesting as they can be quite challenging to solve.

Since $\nabla^2 F(y) = Q + 1000 \overline{A}^T S \overline{A}$ where $S$ is diagonal with entries $(\overline{b} - \overline{A} y)_i^{-2}$, these problems

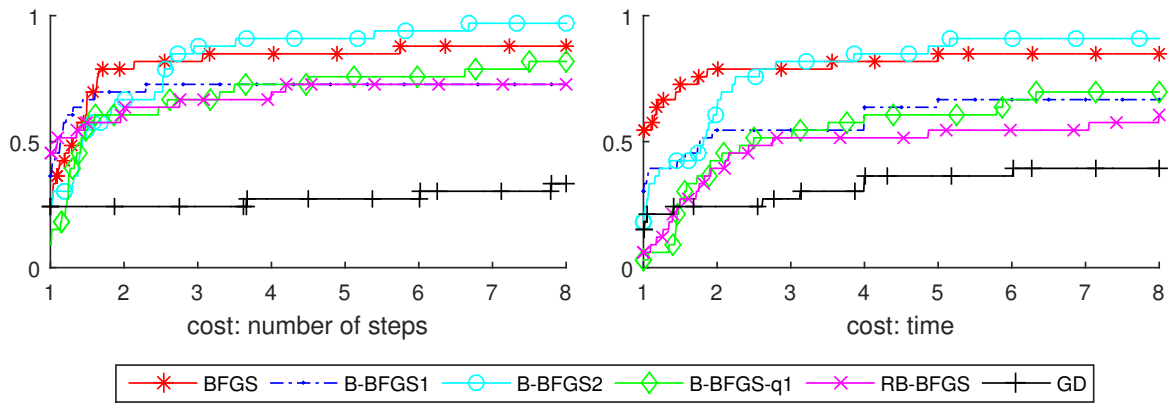


Figure 2.2: Log Barrier QP profiles ($\rho_s(r)$)

are often extremely ill-conditioned. This leads to issues when using `WolfeLineSearch`, as the line search can require many backtracking iterations, or even fail completely, when the current point is near the boundary of the log barrier. This causes particular issues with block updates, as $\nabla^2 F(y)$ has small numerical rank when $S$ has a small number of extremely large entries. Consequently, we removed problems from the test set which were ill-conditioned to the extent that even after performing step filtering, the line search failed at some step before reaching the optimal solution. Quasi-Newton methods, and those using block updates with large $q$ in particular, are poorly suited for these ill-conditioned problems. However, although the standard BFGS method also can fail on these problems, it is more robust than block methods.

### 2.7.2 Non-Convex Experiments

Since non-convex functions often have multiple stationary points, more complex behavior is possible than in the convex case. For instance, one algorithm may generally require more steps to converge, but may be taking advantage of additional information to help avoid spurious local minima.

Let $f_p$ denote the best objective value obtained for problem $p$ by any algorithm. To evaluate both the early and asymptotic performance of our algorithms, we generated performance profiles comparing the cost for each algorithm to reach a solution with objective value less than $f_p + \epsilon|f_p|$ for $\epsilon = 0.2$, $\epsilon = 0.1$, and $\epsilon = 0.01$. When $|f_p|$ is very small (for instance, $|f_p| < 10^{-10}$), we

essentially have $f_p = 0$ and treat all solutions with objective value within $10^{-10}$ as being optimal.

We compared four different algorithms for non-convex minimization:

1. *Damped BFGS*, or *D-BFGS*

   Damped BFGS with $\phi = 0.2$ (see Section 2.6).

2. *Block BFGS*, or *B-BFGS*

   Block BFGS (Algorithm 1) with $q = \lfloor n^{1/3} \rfloor$ and $\tau = 10^{-5}$.

3. *Block BFGS with $q = 1$*, or *B-BFGS-q1*

   Block BFGS (Algorithm 1) with $q = 1$ and $\tau = 10^{-5}$.

4. *Gradient Descent*, or *GD*

**Hyperbolic Tangent Loss Tests**

This is also a classification technique; however, unlike the logistic regression problems in Section 2.7.1, these problems are generally non-convex. Given a set of $m$ data points $(y_i, x_i)$ where $y_i \in \{0, 1\}$ is the class, and $x_i \in \mathbb{R}^n$ the features, we seek to minimize over $w \in \mathbb{R}^n$ the loss function

$$L(w) = \frac{1}{m} \sum_{i=1}^{m} \big(1 - \tanh(y_i x_i^T w)\big) + \frac{1}{2m} \|w\|^2$$

Figure 2.3 presents performance profiles for $\epsilon = 0.2, 0.1, 0.01$, with cost measured in both steps and CPU time. See Appendix 2.9 for a list of the data sets.

B-BFGS and gradient descent perform well at first, making rapid progress to within $0.2|f_p|$ of $f_p$ in the fewest number of steps. B-BFGS continues to converge quickly, generally requiring the fewest steps to reach $0.1|f_p|$ and $0.01|f_p|$ of $f_p$, while gradient descent is overtaken by BFGS and B-BFGS-q1.

Surprisingly, all four algorithms used nearly the same amount of CPU time, with each algorithm completing a majority of problems after using only 1% more time than the fastest algorithm.

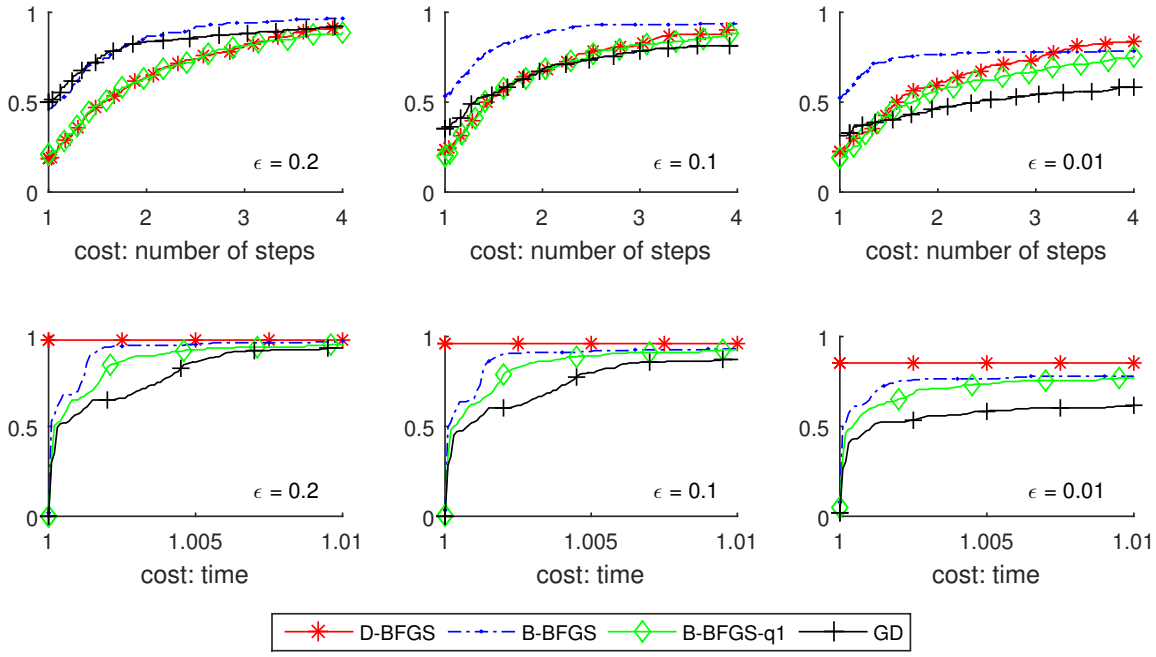


Figure 2.3: Hyperbolic Tangent Loss profiles ($\rho_s(r)$)

**Standard Benchmark Tests**

This test used 19 functions from the test collection of Andrei [50], many of which originate from the CUTEst test set. The functions are listed below, with the number of variables $n$ in parentheses:

`arwhead` (300), `bdqrtic` (200), `cube` (400), `diag1` (250), `dixonprice` (200), `edensch` (300), `eg2` (400), `explin2` (200), `fletchcr` (400), `genhumps` (250), `indef` (250), `mccormick` (400), `raydan1` (400), `rosenbrock` (300), `sine` (400), `sinquad` (400), `tointgss` (200), `trid` (200), `whiteholst` (300).

The gradients and Hessians were computed using the automatic differentiation program ADiGator [51].

For each of these functions, we generated 6 random starting points and tested the 4 algorithms using each starting point, for a total of 114 problems. Figure 2.4 presents performance profiles for $\epsilon = 0.2, 0.1, 0.01$, with cost measured in steps. We see from Figure 2.4 that D-BFGS consistently outperforms B-BFGS-q1, which suggests that Powell's damping method is superior to cautious updates.

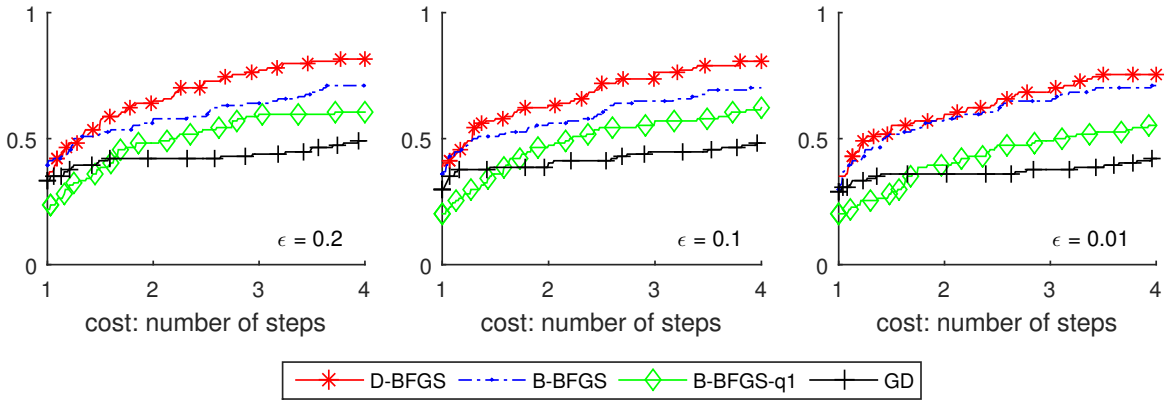


Figure 2.4: Standard Benchmark profiles ($\rho_s(r)$)

## 2.8 Concluding Remarks

We have shown that Block BFGS provides the same theoretical rate of convergence as the classical BFGS method. Further investigation is needed to determine how Block BFGS performs on a wider range of real problems. In our experiments, we focused on a very basic implementation of Block BFGS, but many simple heuristics for improving performance and numerical stability are possible. In particular, it is important to select good values of $q$ and $\tau$ based on insights from the problem domain. We also briefly investigated the effect of using the action of the Hessian on the previous step versus the change in gradient over the previous step (as in classical BFGS) in constructing the update. Further study of the benefits and drawbacks of such an approach would be of interest, as would study of parallel implementation. We hope that this work will serve as a useful foundation for future research on quasi-Newton methods using block updates.

## 2.9 Supplementary: Details of Experiments

### 2.9.1 Logistic Regression Tests (2.7.1)

The following 18 data sets from LIBSVM [52] were used:

`a1a`, `a2a`, `a3a`, `a4a`, `australian`, `colon-cancer`, `covtype`, `diabetes`, `duke`, `ionosphere-scale`, `madelon`, `mushrooms`, `sonar-scale`, `splice`, `svmguide3`, `w1a`, `w2a`, `w3a`.

Each data set was partitioned into 3 disjoint subsets with at most 2000 points. For each subset,

we have a problem of the form (2.7.1) with the standard $L_2$ regularizer $Q = I$, producing 54 standard problems. An additional 96 problems with $Q = I + Q'$ were produced by adding a randomly generated convex quadratic $Q'$ to one of the standard problems. Two such problems were produced for each standard problem, except those from `duke` and `colon-cancer` (omitted for problem size).

### 2.9.2 Log Barrier QP Tests (2.7.1)

Given a convex quadratic program $\min_{x \in \mathbb{R}^n} \{\frac{1}{2} x^T Q x + c^T x \mid Ax = b, x \geq 0\}$, we derive a log barrier QP problem as follows. Taking a basis $N$ for the null space of $A$ (of dimension $s$), and a solution $Ax_0 = b, x_0 \geq 0$, the given QP is equivalent to $\min_{y \in \mathbb{R}^s} \{\frac{1}{2} y^T \overline{Q} y + \overline{c}^T y \mid \overline{A} y \leq \overline{b}\}$, where $\overline{Q} = N^T Q N, \overline{c} = N^T (c + Q x_0), \overline{b} = x_0$ and $\overline{A} = -N$. Replacing the constraint by a log barrier $-\mu \sum_{i=1}^n \log(\overline{b} - \overline{A} y)_i$ (with $\mu = 1000$), we obtain problem (2.7.2).

This test included 43 problems in total. There were 35 log barrier problems derived from the QP test collection of Maros and Mészáros [53]:
`cvxqp1_m`, `cvxqp1_s`, `cvxqp2_m`, `cvxqp2_s`, `cvxqp3_m`, `cvxqp3_s`, `dual1`, `dual2`, `dual3`, `dual4`, `primal1`, `primal3`, `primal4`, `primalc1`, `primalc2`, `primalc5`, `primalc8`, `q25fv47`, `qbeaconf`, `qgrow15`, `qgrow22`, `qgrow7`, `qisrael`, `qscagr7`, `qscfxm1`, `qscfxm2`, `qscfxm3`, `qscorpio`, `qscrs8`, `qsctap1`, `qsctap3`, `qshare1b`, `qship08l`, `stadat1`, `stadat2`.

An additional 8 problems were derived from the following LP problems in the COAP collection [54]: `adlittle`, `agg`, `agg2`, `agg3`, `bnl1`, `brandy`, `fffff800`, `ganges`.

### 2.9.3 Hyperbolic Tangent Loss Tests (2.7.2)

This test used the same data sets as the logistic regression test, with `duke` omitted because of large problem size ($n = 7130$). As in the logistic regression test, each data set was partitioned into 3 subsets with at most 2000 points, producing 51 loss functions. For each loss function, we tried 4 random starting points, for a total of 204 problems.

# Chapter 3: Superlinear Convergence Without Line Searches for Self-Concordant Functions

## 3.1 Introduction

We are concerned in this paper with iterative optimization algorithms, which at each step, first select a *direction* $d_k$ and then determine a *step size* $t_k$. Such algorithms, which are usually referred to as *line search* algorithms, need to choose an appropriate step size $t_k$ to perform well, both in theory and in practice.

Theoretical proofs of global convergence generally assume one of the following approaches for selecting the step sizes:

1. The step sizes are obtained from line searches.

2. The step size is a constant, often chosen 'sufficiently small'.

Inexact line searches, and in particular those that choose steps that satisfy the Armijo-Wolfe conditions, or just the latter combined with backtracking, are usually performed and work well in practice. However, they can be costly to perform, and are often prohibitively costly for many common objective functions such as those that arise in machine learning, computer vision, and natural language processing. Moreover, in stochastic optimization algorithms, line searches based on stochastic function values and gradients, which are only estimates of the true quantities (see Section 3.8), can be meaningless. In contrast, constant step sizes $t_k = t$ for all $k$ require no additional computation beyond selecting $t$, but determining an appropriate constant $t$ may be difficult. The value of $t$ required in the theoretical analysis is often too small for practical purposes, and moreover, is impossible to compute without knowledge of unknown parameters (e.g. the Lipschitz constant of $\nabla f$). A single constant step size may also be highly suboptimal, as the iterates

transition between regions with different curvature.

The basic idea for a step size determined by the local curvature of the objective function $f$ was developed by Nesterov, who introduced the *damped Newton method* [33]. This idea is closely related to a well-behaved class of functions known as *self-concordant functions* [55], which we define in Section 3.3. When applied to a self-concordant function $f$, the damped Newton method is globally convergent and locally converges quadratically. These results were extended in recent work.

1. Tran-Dinh et al. [56] propose a proximal framework for composite self-concordant minimization, which includes proximal damped Newton, proximal quasi-Newton, and proximal gradient descent. They establish that proximal damped Newton is globally convergent and locally quadratically convergent, and that proximal damped gradient descent is globally convergent and locally linearly convergent. However, they do not propose a proximal quasi-Newton algorithm or prove global convergence for a generic version of such an algorithm.

2. Zhang and Xiao [57] propose a distributed method for self-concordant empirical loss functions, based on the damped Newton method, and establish its convergence.

3. Lu [58] proposes a randomized block proximal damped Newton method for composite self-concordant minimization, and establishes its convergence.

While the damped Newton method has been extensively studied, no comparable theory exists for quasi-Newton methods in the self-concordant setting. It is well known that for convex functions, proving global convergence for the BFGS method [59, 25, 23, 26] with inexact line searches is far more challenging than proving global convergence for scaled gradient methods, and that a similar statement holds for the $Q$-superlinear convergence of the BFGS method applied to strongly convex functions compared with, for example, proving $Q$-quadratic convergence of Newton's method. With regard to $Q$-superlinear convergence, it is well known [41] that if the the largest eigenvalue of the Hessian of the objective is bounded above, and if the sum of the distances of the iterates generated by the BFGS method from the global minimizer is finite, then the BFGS

method converges $Q$-superlinearly. We note that Tran-Dinh et al. [56] give a proof of this local result for their "pure"-proximal-BFGS method (i.e., one that uses a step size of 1 on every iteration and starts from a point "close" to the global minimizer), but they do not prove that this method generates iterates satisfying the required conditions. This leaves open the question of how to design a globally convergent "dampled" version of the BFGS method for self-concordant functions. In particular, we wish to avoid assuming either the Dennis-Moré condition [43] or the summability of the distances to the global minmizer, since these conditions are extremely strong, verging on being tautological, as assumptions.

In this paper we extend the theory of self-concordant minimization developed by Nesterov and Nemirovski [55] and further developed by Tran-Dinh et al. [56]. Our focus here is mainly on filling the gap in this theory for quasi-Newton methods. To simplify the presentation, we consider only quasi-Newton methods that use the BFGS update, although our results apply to all methods in the Broyden class of quasi-Newton methods other than the DFP method [28, 29]. We introduce a framework for non-composite optimization; i.e., we do not consider proximal methods as in [56]. The key feature of this framework is a step size that is optimal with respect to an upper bound on the decrease in the objective value, which we call the *curvature-adaptive step size*. We use the term curvature-adaptive, or simply *adaptive*, to refer to this step size choice or to methods that employ it, so as not to confuse such methods with damped BFGS updating methods (e.g., see [60, §18.3]), which are unrelated.

We first prove that scaled gradient methods that use the curvature-adaptive step size are globally $R$-linearly convergent on strongly convex self-concordant functions. We note that in [56], this step size is also identified, but that the $R$-linear convergence is only proved locally. We then prove our main result, on quasi-Newton methods: that the BFGS method, using this step size, is globally convergent for functions that are self-concordant, bounded below, and have a bounded Hessian, and furthermore, is $Q$-superlinearly convergent when the function is strongly convex and self-concordant. For completeness, we then present several numerical experiments which shed insight on the behavior of adaptive methods. These show that for deterministic optimization, using

curvature-based step sizes in quasi-Newton methods is dominated by using inexact line searches, whereas in stochastic settings, using curvature-based step sizes is very helpful compared to constant step sizes.

Our paper is organized as follows. In Section 3.2, we introduce the notation and assumptions that we use throughout the paper. In Section 3.3, we define the class of self-concordant functions and describe their essential properties. In Section 3.4, we introduce our framework for self-concordant minimization and provide a derivation of what we call the *curvature-adaptive* step size, which extends the curvature-based step size obtained in [56] for proximal gradient methods. In Section 3.5, we apply our approach to scaled gradient methods, and give a simple proof that these methods are globally $R$-linearly convergent on strongly convex self-concordant functions. In Section 3.6, we present our main results. Specifically, we prove there that the BFGS method with curvature-adaptive step sizes is globally and $Q$-superlinearly convergent. In Section 3.8, we discuss stochastic extensions of adaptive methods. In Section 3.9.1, we present numerical experiments testing our new methods on logistic regression problems in the deterministic setting. In Section 3.9.2, we provide a numerical example of solving an online stochastic problem using stochastic adaptive methods.

### 3.2 Preliminaries

We use $f : \mathbb{R}^n \to \mathbb{R}$ to denote the objective function, and $g(\cdot), G(\cdot)$ denote the gradient $\nabla f(\cdot)$ and Hessian $\nabla^2 f(\cdot)$, respectively. In the context of a sequence of points $\{x_k\}_{k=0}^{\infty}$, we write $g_k$ for $g(x_k)$ and $G_k$ for $G(x_k)$. Unless stated otherwise, the function $f$ is assumed to have continuous third derivatives (as $f$ is generally assumed to be self-concordant), which we write as $f \in \mathcal{C}^3$.

The norm $\|\cdot\|$ denotes the 2-norm, and when applied to a matrix, the operator 2-norm.

### 3.3 Self-Concordant Functions

The notion of *self-concordant* functions was first introduced by Nesterov and Nemirovski [55] for their analysis of Newton's method in the context of interior-point methods. Nesterov [33]

provides a clear exposition and motivates self-concordancy by observing that, while Newton's method is invariant under affine transformations, the convergence analysis makes use of norms which are *not* invariant. To remedy this, Nesterov and Nemirovski replace the Euclidean norm by an invariant local norm, and replace the assumption of Lipschitz continuity of the Hessian $G(x)$ by the self-concordancy of $f$.

**Definition.** *Let $f$ be a convex function. The* local norm *of $h \in \mathbb{R}^n$ at a point $x$ where $G(x) \succ 0$ is given by*

$$\|h\|_x = \sqrt{h^T G(x) h}.$$

**Definition.** *A closed convex function $f : \mathbb{R}^n \to \mathbb{R}$ is* self-concordant *if $f \in \mathcal{C}^3$ and there exists a constant $\kappa$ such that for every $x \in \mathbb{R}^n$ and every $h \in \mathbb{R}^n$, we have*

$$|\nabla^3 f(x)[h,h,h]| \leq \kappa (\nabla^2 f(x)[h,h])^{3/2}.$$

*If $\kappa = 2$, $f$ is* standard self-concordant. *Any self-concordant function can be scaled to be standard self-concordant; the scaled function $\frac{1}{4}\kappa^2 f$ is standard self-concordant. Hence, we assume all self-concordant functions have $\kappa = 2$, unless stated otherwise.*

There is also an equivalent definition which is frequently useful.

**Theorem 3.3.1** (Lemma 4.1.2, [33])**.** *A closed convex function $f$ is self-concordant if and only if for every $x \in \mathbb{R}^n$ and all $u_1, u_2, u_3 \in \mathbb{R}^n$, we have*

$$|\nabla^3 f(x)[u_1,u_2,u_3]| \leq 2 \prod_{i=1}^{3} \|u_i\|_x.$$

The next inequalities are fundamental for self-concordant functions. These results are well known (see [33, §4.1.4]), but for completeness, we provide a proof.

**Lemma 3.3.2.** *Let $f$ be standard self-concordant and strictly convex, and let $x \in \mathbb{R}^n$ and $0 \neq d \in$*

$\mathbb{R}^n$. *Let* $\delta = \|d\|_x$. *Then for all* $t \geq 0$,

$$f(x+td) \geq f(x) + tg(x)^T d + \delta t - \log(1+\delta t) \tag{3.3.1}$$

*and*

$$g(x+td)^T d \geq g(x)^T d + \frac{\delta^2 t}{1+\delta t}. \tag{3.3.2}$$

*For all* $0 \leq t < \frac{1}{\delta}$,

$$f(x+td) \leq f(x) + tg(x)^T d - \delta t - \log(1-\delta t) \tag{3.3.3}$$

*and*

$$g(x+td)^T d \leq g(x)^T d + \frac{\delta^2 t}{1-\delta t}. \tag{3.3.4}$$

*Proof.* Define $\phi : \mathbb{R} \to \mathbb{R}$ by $\phi(t) = d^T \nabla^2 f(x+td)d$. Since $f$ has continuous third derivatives, $\phi(t)$ is continuously differentiable and from the definition of self-concordancy, its derivative satisfies

$$|\phi'(t)| = |\nabla^3 f(x+td)[d,d,d]| \leq 2(\nabla^2 f(x+td)[d,d])^{3/2} = 2\phi(t)^{3/2}. \tag{3.3.5}$$

Moreover, since $f$ is strictly convex and $d \neq 0$, $\phi(t) > 0$ for all $t$. Therefore, from (3.3.5),

$$|\frac{d}{dt}\phi(t)^{-1/2}| = \frac{1}{2}|\phi(t)^{-3/2}\phi'(t)| \leq 1.$$

Defining $\psi(s) = \frac{d}{dt}\phi(t)^{-1/2}\big|_{t=s}$, the above inequality is equivalent to $|\psi(s)| \leq 1$. By Taylor's Theorem, there exists a point $u \in (0,t)$ such that $\phi(t)^{-1/2} - \phi(0)^{-1/2} = t\psi(u)$. Since $|\psi(u)| \leq 1$, we deduce that

$$\phi(0)^{-1/2} - t \leq \phi(t)^{-1/2} \leq \phi(0)^{-1/2} + t.$$

Note that $\delta = \phi(0)^{1/2}$. Rearranging the upper bound, we find that for all $t \geq 0$,

$$\phi(t) \geq \frac{\delta^2}{(1+\delta t)^2}. \tag{3.3.6}$$

Similarly, we find that for $0 \leq t < \frac{1}{\delta}$,

$$\phi(t) \leq \frac{\delta^2}{(1-\delta t)^2}. \tag{3.3.7}$$

Integrating (3.3.6) yields the inequalities (3.3.1), (3.3.2), and integrating (3.3.7) produces (3.3.3), (3.3.4). □

### 3.4 Curvature-Adaptive Step Sizes

We define a general framework for an iterative method with step sizes determined by the local curvature. At each step, we compute a descent direction $d_k = -H_k g_k$, where $H_k$ is a positive definite matrix, and a step size

$$t_k = \frac{\rho_k}{(\rho_k + \delta_k)\delta_k},$$

where

$$\delta_k = \|d_k\|_{x_k}$$

and

$$\rho_k = g_k^T H_k g_k.$$

We then advance to the point $x_{k+1} = x_k + t_k d_k$.

We will refer to the above step size $t_k$ as the *curvature-adaptive* step size, or simply the *adaptive* step size. A method within our framework will be referred to as an *adaptive* method. A generic method in this framework is specified in Algorithm 3.

Note that this framework encompasses several classical methods. When $H_k = I$ for all $k$, the resulting method is gradient descent. When $H_k = G_k^{-1}$, we recover the *damped Newton method* proposed by Nesterov. When $H_k$ is an approximation of $G_k^{-1}$ obtained by applying a quasi-Newton updating formula, the resulting method is a quasi-Newton method. In particular, we will focus on the case where $H_k$ evolves according to the BFGS update formula. We also note that in all variants other than the damped Newton method, we do not access the full Hessian matrix $G_k$ at any step, but

**Algorithm 3** Adaptive Iterative Method

**input**: $x_0, H_0$, variant
**for** $k = 0, 1, 2, \ldots$ **do**
    Set $d_k \leftarrow -H_k g_k$
    Set $\rho_k \leftarrow -g_k^T d_k$
    Set $\delta_k^2 \leftarrow d_k^T G_k d_k$
    Set $t_k \leftarrow \frac{\rho_k}{(\rho_k + \delta_k)\delta_k}$
    Set $x_{k+1} \leftarrow x_k + t_k d_k$
    **if** variant (i): gradient descent **then**
        $H_{k+1} \leftarrow I$
    **end if**
    **if** variant (ii): Newton **then**
        $H_{k+1} = G_{k+1}^{-1}$
    **end if**
    **if** variant (iii): BFGS **then**
        Use standard BFGS formula (3.6.1) to compute $H_{k+1}$
    **end if**
    **if** variant (iv): L-BFGS **then**
        Update L-BFGS curvature pairs
    **end if**
**end for**

only the action of $G_k$ on the direction $d_k$, which typically requires a computational effort similar to that required to compute the gradient $g_k$.

Using the results of Section 3.3, we now show that the curvature-adaptive step size $t_k = \frac{\rho_k}{(\rho_k + \delta_k)\delta_k}$ in Algorithm 3 maximizes a lower bound on the decrease in $f$ obtained by taking a step in the direction $d_k$.

**Lemma 3.4.1.** *Suppose $f$ is self-concordant and strictly convex. At iteration $k$ of Algorithm 3, taking the step $t_k d_k$, where $d_k = -H_k g_k$ and $t_k = \frac{\rho_k}{(\rho_k + \delta_k)\delta_k}$, yields the point $x_{k+1} = x_k + t_k d_k$ at which the objective function $f(x_{k+1})$ satisfies*

$$f(x_{k+1}) \leq f(x_k) - \omega(\eta_k) \tag{3.4.1}$$

*where*

$$\eta_k = \frac{\rho_k}{\delta_k}$$

*and* $\omega : \mathbb{R} \to \mathbb{R}$ *is the function* $\omega(z) = z - \log(1 + z)$.

*Moreover, the step size* $t_k$ *minimizes the upper bound (3.3.3) on* $f(x_{k+1})$ *provided by Lemma 3.3.2.*

*Proof.* We fix the index $k$ and omit the subscripts for brevity. First, observe that

$$0 \leq t = \frac{\rho}{(\rho + \delta)\delta} < \frac{1}{\delta}.$$

Therefore, we can apply inequality (3.3.3) to $f(x + td)$. Noting that $\rho = -g^T d$, (3.3.3) can be written as $f(x + td) \leq f(x) - \Delta(t)$ where $\Delta(\cdot)$ is defined to be the function $\Delta(\tau) = (\rho + \delta)\tau + \log(1 - \delta\tau)$. For the curvature-adaptive step size $t$, it is easily verified that

$$\Delta(t) = \Delta\left(\frac{\rho}{(\rho + \delta)\delta}\right) = \frac{\rho}{\delta} + \log\left(\frac{\delta}{\rho + \delta}\right) = \frac{\rho}{\delta} - \log\left(1 + \frac{\rho}{\delta}\right) = \omega(\eta).$$

Furthermore, for $0 \leq \tau < \frac{1}{\delta}$, $\frac{d}{d\tau}\Delta(\tau) = \rho + \delta - \frac{\delta}{1-\delta\tau}$ and $\frac{d^2}{d\tau^2}\Delta(\tau) = -\frac{\delta^2}{(1-\delta\tau)^2}$. We find that $\frac{d}{d\tau}\Delta(t) = 0$ and $\frac{d^2}{d\tau^2}\Delta(t) \leq 0$, which implies that $\Delta$ is maximized at $\tau = t = \frac{\rho}{(\rho+\delta)\delta}$. □

Since $\omega(\eta) = \eta - \log(1 + \eta)$ is positive for all $\eta > 0$, it follows that if $\limsup_k \eta_k > 0$, then $f(x_k) \to -\infty$. This simple fact will be crucial in our convergence analysis.

**Lemma 3.4.2.** *If, in addition to the assumptions in Lemma 3.4.1,* $f$ *is bounded below, then* $\eta_k = \frac{\rho_k}{\delta_k} \to 0$ *for any of the adaptive variants in Algorithm 3.*

*Proof.* By Lemma 3.4.1, $f(x_k)$ satisfies $f(x_k) \leq f(x_0) - \sum_{j=0}^{k-1} \omega(\eta_j)$. Suppose that $\limsup_k \eta_k > 0$. Since the function $\omega(\eta)$ is positive and monotonically increasing for $\eta > 0$, we have $\limsup_k \omega(\eta_k) = \omega(\limsup_k \eta_k) > 0$. Hence $f(x_k) \to -\infty$, a contradiction. □

In terms of $g_k$, $H_k$, and $G_k$, the adaptive step size $t_k$ can be expressed as

$$t_k = \frac{g_k^T H_k g_k}{g_k^T H_k G_k H_k g_k + g_k^T H_k g_k \sqrt{g_k^T H_k G_k H_k g_k}}.$$

This formula relates $t_k$ to the local curvature. When the curvature of $f$ in the direction $d_k = -H_k g_k$ is relatively flat, the local norm $\|d_k\|_{x_k} = \sqrt{g_k^T H_k G_k H_k g_k}$ is small, and the adaptive step size $t_k$

will be large. Conversely, when the curvature of $f$ in the direction $d_k$ is steep, $t_k$ will be small. Intuitively, this is precisely the desired behavior for a step size, since we wish to take larger steps when the function changes slowly, and smaller steps when the function changes rapidly.

### 3.5 Scaled Gradient Methods

We first consider the class of methods where the matrices $H_k$ are positive definite and uniformly bounded above and below. That is, there exist positive constants $\lambda, \Lambda$ such that for every $k \geq 0$,

$$\lambda I \preceq H_k \preceq \Lambda I. \tag{3.5.1}$$

The convergence analysis is rather straightforward, as seen in the proofs of the following two theorems for these methods.

**Theorem 3.5.1.** *If $f$ is self-concordant, strictly convex, bounded below, and the Hessian satisfies $G(x) \preceq MI$ on the level set $\Omega = \{x : f(x) \leq f(x_0)\}$, then any adaptive method (Algorithm 3) for which the matrices $H_k$ satisfy equation* (3.5.1) *converges globally in the sense that $\lim_{k\to\infty} \|g_k\| = 0$.*

*Proof.* Since $H_k$ is positive definite, $H_k^{1/2}$ exists and we may define $z_k = H_k^{1/2} g_k$. Observe that

$$\eta_k = \frac{g_k^T H_k g_k}{\sqrt{g_k^T H_k G_k H_k g_k}} = \frac{z_k^T z_k}{\sqrt{z_k^T (H_k^{1/2} G_k H_k^{1/2}) z_k}} \geq \frac{\|z_k\|}{\sqrt{\Lambda M}} \geq \sqrt{\frac{\lambda}{\Lambda M}} \|g_k\| \tag{3.5.2}$$

where we have used the fact that the maximum eigenvalue of $H_k^{1/2} G_k H_k^{1/2}$ is bounded by $\Lambda M$. By Lemma 3.4.2, $\eta_k \to 0$. Therefore $\|g_k\| \to 0$. □

If in addition, $f$ is strongly convex with $mI \preceq G(x)$ for $m > 0$, then an adaptive method satisfying equation (3.5.1) is globally $R$-linearly convergent. The proof uses the fact that strongly convex functions satisfy the Polyak-Łojasiewicz inequality (Lemma 2.5.3).

We are now ready to prove the $R$-linear convergence of adaptive scaled gradient methods.

**Theorem 3.5.2.** *If $f$ is self-concordant and strongly convex (so there exist constants $0 < m \leq M$ such that $mI \preceq G(x) \preceq MI$ for all $x \in \Omega$), then an adaptive method (Algorithm 3) for which the matrices $H_k$ satisfy equation* (3.5.1) *is globally $R$-linearly convergent. That is, there exists a positive constant $\gamma < 1$ such that $f(x_{k+1}) - f(x_*) \leq \gamma(f(x_k) - f(x_*))$ for all $k$.*

*Proof.* Since $\eta_k \to 0$ by Lemma 3.4.2, the sequence $\{\eta_k\}_{k=0}^{\infty}$ is bounded. Let $\Gamma = \sup_k \eta_k < \infty$, and let $c = \frac{1}{2(1+\Gamma)}$. Observe that $\omega(z) = z - \log(1+z) \geq cz^2$ for $0 \leq z \leq \Gamma$, as $\omega(0) = 0$ and $\frac{d}{dz}(\omega(z) - cz^2) = \frac{z(1-2c-2cz)}{1+z}$, which is non-negative for $0 \leq z \leq \Gamma$. Hence, since $\eta_k \leq \Gamma$ for all $k$, we have

$$\begin{aligned} f(x_{k+1}) - f(x_*) \leq f(x_k) - f(x_*) - \omega(\eta_k) &\leq f(x_k) - f(x_*) - c\eta_k^2 \\ &\leq f(x_k) - f(x_*) - \frac{c\lambda}{\Lambda M}\|g(x_k)\|^2 \\ &\leq \left(1 - \frac{\lambda m}{\Lambda(1+\Gamma)M}\right)(f(x_k) - f(x_*)) \end{aligned}$$

where the first line follows from inequality (3.3.3), the second from inequality (3.5.2), and the third from Lemma 2.5.3. Taking $\gamma = 1 - \frac{\lambda m}{\Lambda(1+\Gamma)M}$, we obtain the desired $R$-linear convergence. □

### 3.5.1 Adaptive Gradient Descent

When $H_k = I$ for all $k$ in Algorithm 3, the method corresponds to gradient descent with adaptive step sizes that incorporate second-order information. This strategy for selecting analytically computable step sizes may have several advantages in practice. Using second-order information allows a better local model of the objective function. The classical analysis of gradient descent with a fixed step size also generally requires a sufficiently small step size in order to guarantee convergence. This step size is a function of the Lipschitz constant for the gradient $g(x)$, which is either unknown or impractical to compute. The step size needed to ensure convergence in theory is also often impractically tiny, leading to slow convergence in practice. For the class of self-concordant functions, an adaptive step size can be easily computed without knowledge of any constants, and still provides a theoretical guarantee of convergence, which is a significant advantage.

A proximal gradient descent method with adaptive step sizes was studied by Tran-Dinh et al. [56], who proved the method to be globally convergent for self-concordant functions, and locally $R$-linearly convergent for strongly convex self-concordant functions. However, our convergence analysis above employs different techniques from those in [56], and in particular, we obtain the following theorem, which shows that the adaptive gradient descent method is globally $R$-linearly convergent, as an immediate corollary of Theorem 3.5.1 and Theorem 3.5.2:

**Theorem 3.5.3.** *Suppose that $f$ is self-concordant, strictly convex, bounded below, and $G(x) \preceq MI$ on the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Then the adaptive gradient descent method converges in the sense that $\lim_{k\to\infty} \|g_k\| = 0$. Furthermore, if $f$ is strongly convex on $\Omega$, then the adaptive gradient descent method is globally $R$-linearly convergent.*

### 3.5.2 Adaptive L-BFGS

The limited-memory BFGS algorithm (L-BFGS, [40]) stores a fixed number of previous *curvature pairs* $(s_k, y_k)$, where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, and computes $d_k = -H_k g_k$ from the curvature pairs using a two-loop recursion [61]. It is well known that L-BFGS satisfies equation (3.5.1). In [31], the following bounds are obtained.

**Theorem 3.5.4** (Lemma 1, [31])**.** *Suppose that $f$ is strongly convex, with $mI \leq G(x) \leq MI$. Let $\ell$ be the number of curvature pairs stored by the L-BFGS method. Then the matrices $H_k$ satisfy*

$$\lambda I \preceq H_k \preceq \Lambda I,$$

*where $\lambda = (1 + \ell M)^{-1}$ and $\Lambda = (1 + \sqrt{\kappa})^{2\ell} \left(1 + \frac{1}{m(2\sqrt{\kappa}+\kappa)}\right)$ for $\kappa = M/m$.*

Hence, it follows immediately from Theorem 3.5.1 and Theorem 3.5.2 that:

**Theorem 3.5.5.** *Suppose that $f$ is self-concordant, strongly convex, and $mI \preceq G(x) \preceq MI$ on the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Then the adaptive L-BFGS method is globally $R$-linearly convergent.*

We note that, as with gradient descent, it is well known that the L-BFGS method converges if inexact Armijo-Wolfe line searches are performed, or a sufficiently small fixed step size, that depends on the Lipschitz constant of $g(x)$, is used.

## 3.6 Adaptive BFGS

If $H_k$ is chosen to approximate $(\nabla^2 f(x_k))^{-1}$, then we obtain quasi-Newton methods with adaptive step sizes. In particular, we may iteratively update $H_k$ using the BFGS update formula, which we briefly describe. Let $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. The BFGS update sets $H_{k+1}$ to be the nearest matrix to $H_k$ (in a variable metric) satisfying the *secant equation* $H_{k+1} y_k = s_k$[23]. It is well known that $H_{k+1}$ has the following expression in terms of $H_k, s_k$ and $y_k$:

$$H_{k+1} = \frac{s_k s_k^T}{y_k^T s_k} + \left( I - \frac{s_k y_k^T}{y_k^T s_k} \right) H_k \left( I - \frac{y_k s_k^T}{y_k^T s_k} \right). \tag{3.6.1}$$

### 3.6.1 Superlinear Convergence of Adaptive BFGS

The convergence analysis of the classical BFGS method [41, 27] assumes that the method uses inexact line searches satisfying the *Armijo-Wolfe* conditions: for constants $c_1, c_2 \in (0, 1)$ with $c_1 < \frac{1}{2}$ and $c_1 < c_2$, the step size $t_k$ should satisfy

$$f(x_k + t_k d_k) \leq f(x_k) + c_1 t_k g_k^T d_k \qquad \text{(Armijo condition)} \tag{3.6.2}$$

and

$$g(x_k + t_k d_k)^T d_k \geq c_2 g_k^T d_k. \qquad \text{(Wolfe condition)} \tag{3.6.3}$$

Under the assumption of Armijo-Wolfe line searches, Powell [41] proves the following global convergence theorem for BFGS.

**Theorem 3.6.1** (Lemma 1, [41])**.** *If the BFGS algorithm with Armijo-Wolfe inexact line search is applied to a convex function $f(x)$ that is bounded below, if $x_0$ is any starting vector and $H_0$ is any*

*positive definite matrix, and if the Hessian $G(x)$ satisfies $G(x) \preceq MI$ for all $x$ in the level set $\Omega = \{x : f(x) \leq f(x_0)\}$, then the limit*

$$\liminf_{k\to\infty} \|g_k\| = 0 \tag{3.6.4}$$

*is obtained.*

In our setting, $f$ is a self-concordant and strictly convex function that is bounded below and satisfies $G(x) \preceq MI$. In order to prove that adaptive BFGS is convergent in the sense of the limit (3.6.4), it suffices to show that the adaptive step sizes $t_k$ satisfy the Armijo condition for any $c_1 < \frac{1}{2}$, and eventually satisfy the Wolfe condition for any $c_2 < 1$ (i.e. there exists some $k_0$ such that the Wolfe condition is satisfied for all $k \geq k_0$). Specifically, we prove the following two theorems that apply to *every* adaptive method described by Algorithm 3.

**Theorem 3.6.2.** *Let $f$ be self-concordant and strictly convex. The curvature-adaptive step size $t_k = \frac{\rho_k}{(\rho_k+\delta_k)\delta_k}$ satisfies the Armijo condition for any $c_1 \leq \frac{1}{2}$.*

*Proof.* Let $c_1 \leq \frac{1}{2}$. We aim to prove that $f(x_{k+1}) \leq f(x_k) + c_1 t_k g_k^T d_k$. By Lemma 3.4.1, $f(x_{k+1}) \leq f(x_k) - \omega(\eta_k)$. Therefore, it suffices to prove that

$$\omega(\eta_k) \geq -\frac{1}{2} t_k g_k^T d_k.$$

For brevity, we omit the index $k$. Notice that

$$-tg^T d = tg^T Hg = t\rho = \frac{\rho^2}{(\rho+\delta)\delta} = \frac{\rho^2/\delta^2}{\rho/\delta + 1} = \frac{\eta^2}{1+\eta}.$$

Therefore, we must prove that for $\eta \geq 0$,

$$\omega(\eta) = \eta - \log(1+\eta) \geq \frac{1}{2}\frac{\eta^2}{1+\eta}.$$

Define $\zeta(z) = z - \log(1+z) - \frac{1}{2}\frac{z^2}{1+z}$. Observe that $\zeta(0) = 0$ and

$$\frac{d}{dz}\zeta(z) = 1 - \frac{1}{1+z} - \frac{1}{2}\frac{z^2+2z}{(1+z)^2} = \frac{1}{2}\frac{z^2}{(1+z)^2}.$$

Since $\frac{d}{dz}\zeta(z) \geq 0$ for all $z \geq 0$, we conclude that $\omega(\eta) \geq \frac{1}{2}\frac{\eta^2}{1+\eta}$ for all $\eta \geq 0$. This completes the proof. □

**Theorem 3.6.3.** *Let $f$ be self-concordant, strictly convex, and bounded below. Suppose that $\{x_k\}_{k=0}^{\infty}$ is a sequence of iterates generated by Algorithm 3. For any $0 < c_2 < 1$, there exists an index $k_0$ such that for all $k \geq k_0$, the curvature-adaptive step size $t_k$ satisfies the Wolfe condition.*

*Proof.* We aim to prove that $g_{k+1}^T d_k \geq c_2 g_k^T d_k$. This is equivalent to $g(x_k + t_k d_k)^T d_k - g(x_k)^T d_k \geq -(1-c_2)g(x_k)^T d_k = (1-c_2)\rho_k$. By inequality (3.3.2) with $\delta = \delta_k$ and $t = t_k$, we have

$$g(x_k + t_k d_k)^T d_k - g(x_k)^T d_k \geq \frac{\delta_k^2 t_k}{1+\delta_k t_k} = \frac{\delta_k \rho_k}{2\rho_k + \delta_k} = \frac{1}{1+2\eta_k}\rho_k. \tag{3.6.5}$$

Since $f$ is bounded below, Lemma 3.4.2 implies that $\eta \to 0$, and hence there exists some $k_0$ such that $\frac{1}{1+2\eta_k} \geq 1 - c_2$ for all $k \geq k_0$. □

Note that the assumption of strict convexity also implies that $y_k^T s_k > 0$, so the BFGS update is well-defined.

We can now immediately apply Theorem 3.6.1 to deduce that adaptive BFGS is convergent. Since there always exists an index $k_0$ such that the Armijo-Wolfe conditions are satisfied for all $k \geq k_0$, we can consider the subsequent iterates $\{x_k\}_{k=k_0}^{\infty}$ as arising from the classical BFGS method with initial matrix $H_{k_0}$.

**Theorem 3.6.4.** *Let $f$ be self-concordant, strictly convex, bounded below, whose Hessian satisfies $G(x) \preceq MI$ for all $x \in \Omega$. Then for the adaptive BFGS method, $\liminf_{k\to\infty} \|g_k\| = 0$.*

It is also possible to directly prove Theorem 3.6.4 by analyzing the evolution of the trace and

determinant of $H_k$, but the resulting proof, which is quite long, does not provide clarity on the essential properties of the adaptive step size.

It is well known that if the objective function $f$ is strongly convex, then the classical BFGS method converges $Q$-superlinearly. Let us now assume that $f$ is strongly convex, so there exist constants $0 < m \leq M$ with $mI \preceq G(x) \preceq MI$ for all $x \in \Omega$. Let $x_*$ denote the unique minimizer of $f$.

**Theorem 3.6.5** (Lemma 4, [41])**.** *Let $f$ be strongly convex, and let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates generated by the BFGS method with inexact Armijo-Wolfe line searches. Then $\sum_{k=0}^{\infty} \|x_k - x_*\| < \infty$.*

Since the adaptive step size $t_k$ eventually satisfies the Armijo-Wolfe conditions, the same holds for BFGS with adaptive step sizes.

**Theorem 3.6.6.** *Let $f$ be self-concordant and strongly convex. The sequence of iterates $\{x_k\}_{k=0}^{\infty}$ produced by adaptive BFGS satisfies $\sum_{k=0}^{\infty} \|x_k - x_*\| < \infty$.*

In the proof of superlinear convergence for the classical BFGS method, it is assumed that the Hessian $G(x)$ is Lipschitz continuous. However, it is unnecessary to make this assumption in our setting, as $G(x)$ is necessarily Lipschitz when $f$ is self-concordant and $G(x)$ is bounded above. This fact is not difficult to establish, but we provide a proof for completeness.

**Theorem 3.6.7.** *If $f$ is self-concordant and satisfies $G(x) \preceq MI$ for all $x \in \Omega$, then $G(x)$ is Lipschitz continuous on $\Omega$, with constant $2M^{3/2}$.*

*Proof.* Let $x, y \in \Omega$, and let $e = x - y$. Let $v \in \mathbb{R}^n$ be any unit vector. By Taylor's Theorem, we have

$$v^T G(x) v = v^T G(y) v + \int_0^1 \nabla^3 f(y + \tau e)[v, v, e] d\tau.$$

Hence, by Theorem 3.3.1,

$$\begin{aligned}|v^T(G(x) - G(y))v| &\leq \int_0^1 |\nabla^3 f(y + \tau e)[v, v, e]| d\tau \\ &\leq 2\int_0^1 v^T G(y + \tau e)v\sqrt{e^T G(y + \tau e)e}\, d\tau \\ &\leq 2\int_0^1 M^{3/2}\|e\| d\tau = 2M^{3/2}\|x - y\|.\end{aligned}$$

Therefore, the eigenvalues of $G(x) - G(y)$ are bounded in norm by $2M^{3/2}\|x - y\|$. It follows that $\|G(x) - G(y)\| \leq 2M^{3/2}\|x - y\|$, so $G(x)$ is Lipschitz continuous. □

It is well known that the BFGS method is invariant under an affine change of coordinates, so we may assume without loss of generality that $G(x_*) = I$. This corresponds to considering the function $\widetilde{f}(\widetilde{x}) = f(G(x_*)^{-1/2}\widetilde{x})$ and performing a change of coordinates $\widetilde{x} = G(x_*)^{1/2}x$. By [33, Theorem 4.1.2], the function $\widetilde{f}$ is also self-concordant, with the same $\kappa$ as for $f$.

To complete the proof of superlinear convergence, we use results established by Dennis and Moré [43] and Griewank and Toint [44]. In [44, §4], Griewank and Toint prove that, given Theorem 3.6.6 and Lipschitz continuity of $G(x)$ (Theorem 3.6.7), the following limit holds:

$$\lim_{k\to\infty} \frac{\|(B_k - I)d_k\|}{\|d_k\|} = 0 \tag{3.6.6}$$

Furthermore, the argument in [44] shows that both $\{\|B_k\|\}_{k=0}^{\infty}$ and $\{\|H_k\|\}_{k=0}^{\infty}$ are bounded. Writing $B_k d_k = -g_k$ and $-d_k = H_k g_k$, and using the fact that $\|d_k\| \leq \|H_k\|\|g_k\| \leq \Gamma\|g_k\|$, where $\Gamma$ is an upper bound on the sequence of norms $\{\|H_k\|\}_{k=0}^{\infty}$, we have an equivalent limit

$$\lim_{k\to\infty} \frac{\|H_k g_k - g_k\|}{\|g_k\|} = 0 \tag{3.6.7}$$

This enables us to prove that the adaptive step sizes $t_k$ converge to 1, which is necessary for superlinear convergence.

**Theorem 3.6.8.** *The curvature-adaptive step sizes $t_k$ in the adaptive BFGS method converge to 1.*

*Proof.* We omit the index $k$ for brevity, and define $u = Hg - g$. Since $t$ can be expressed as $t = \frac{\eta/\delta}{1+\eta}$, and we have from Lemma 3.4.2 that $\eta \to 0$, it suffices to show that $\frac{\eta}{\delta}$ converges to 1.

$$\begin{aligned}\frac{\eta}{\delta} = \frac{\rho}{\delta^2} &= \frac{g^T H g}{g^T H G H g} \\ &= \frac{g^T g + g^T u}{g^T G g + 2 g^T G u + u^T G u} \\ &= \frac{1 + \frac{g^T u}{g^T g}}{\frac{g^T G g}{g^T g} + 2\frac{g^T G u}{g^T g} + \frac{u^T G u}{g^T g}}\end{aligned}$$

The limit (3.6.7) implies that $\frac{\|u\|}{\|g\|} \to 0$. Hence, the Cauchy-Schwarz inequality and the upper bound $G(x) \preceq MI$ imply that $\frac{g^T u}{g^T g}, \frac{g^T G u}{g^T g}, \frac{u^T G u}{g^T g}$ converge to 0. Since $G = G(x_k)$ and $x_k \to x_*$, we have $G \to I$, and therefore $\frac{g^T G g}{g^T g} \to 1$. It follows that $\frac{\eta}{\delta}$, and therefore $t$, converges to 1. □

We now make a slight modification to the Dennis-Moré characterization of superlinear convergence. Using the triangle inequality twice and the fact that $G(x_*) = I$, we obtain

$$\begin{aligned}\frac{\|(B_k - I)s_k)\|}{\|s_k\|} &= \frac{\|t_k g_{k+1} - t_k g_k - G(x_*)s_k - t_k g_{k+1}\|}{\|s_k\|} \\ &\geq t_k \frac{\|g_{k+1}\|}{\|s_k\|} - \frac{\|t_k g_{k+1} - t_k g_k - t_k G(x_*)s_k - (1-t_k)G(x_*)s_k\|}{\|s_k\|} \\ &\geq t_k \frac{\|g_{k+1}\|}{\|s_k\|} - \frac{t_k \|\int_0^1 (G(x_k + \tau s_k) - G(x_*))s_k d\tau\|}{\|s_k\|} - |1 - t_k| \frac{\|G(x_*)s_k\|}{\|s_k\|}.\end{aligned}$$

Rearranging, and applying Theorem 3.6.7,

$$\frac{\|g_{k+1}\|}{\|s_k\|} \leq \frac{1}{t_k}\frac{\|(B_k - I)s_k)\|}{\|s_k\|} + 2M^{3/2}\max\{\|x_k - x_*\|, \|x_{k+1} - x_*\|\} + \frac{|1-t_k|}{t_k}M. \quad (3.6.8)$$

Since $x_k \to x_*$ by Theorem 3.6.4 and $t_k \to 1$ by Theorem 3.6.8, both of the latter terms converge to 0. Finally, equation (3.6.6) implies that $\frac{\|(B_k - I)s_k\|}{\|s_k\|}$ converges to 0, so it follows from equation (3.6.8) that $\frac{\|g_{k+1}\|}{\|s_k\|}$ converges to 0.

Since $f$ is strongly convex, $\|g(x)\| \geq m\|x - x_*\|$. Hence, we find that

$$\frac{\|g_{k+1}\|}{\|s_k\|} \geq \frac{m\|x_{k+1} - x_*\|}{\|x_{k+1} - x_*\| + \|x_k - x_*\|},$$

which implies that $\frac{\|x_{k+1}-x_*\|}{\|x_k-x_*\|} \to 0$. Thus, we have the following:

**Theorem 3.6.9.** *Suppose that $f$ is self-concordant, and strongly convex on $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Then the adaptive BFGS method converges $Q$-superlinearly.*

By the same reasoning, the results in [27] and [44] imply that these convergence theorems also hold for the adaptive versions of the quasi-Newton methods in *Broyden's convex class*, with the exception of the DFP method. The adaptive versions of the Block BFGS methods proposed in [1] can also be shown to be $Q$-superlinearly convergent.

### 3.7 Hybrid Step Selection

Consider the damped Newton method of Nesterov, which is obtained by setting $H_k = G_k^{-1}$. This yields $\rho_k = g_k^T G_k^{-1} g_k$ and $\delta_k = \sqrt{g_k G_k^{-1} G_k G_k^{-1} g_k} = \sqrt{\rho}$, whence $\eta = \rho/\delta = \delta$. The curvature-adaptive step size $t$ then reduces to

$$t = \frac{\eta/\delta}{1+\eta} = \frac{1}{1+\delta}.$$

When $\delta$ is large (for example, if the initial point $x_0$ is chosen poorly), then the curvature-adaptive step size may be very small, even when the inverse Hessian approximation $H_k$ is good. This conservatism is the price of the curvature-adaptive step size guaranteeing global convergence (in contrast to Newton's method, which is *not* globally convergent, and to gradient descent, which may diverge if the step size is too large). A small step $t_k d_k$ is likely to result in $t_{k+1}$ also being small[1]. Thus, when the initial $\delta$ is large, a method using adaptive step sizes may produce a long succession of small steps. This suggests the following heuristic for selecting step sizes:

1. Select a set $T_k$ of candidate step sizes for $t_k$.

2. At step $k$, test the elements of $T_k$ in order until a candidate step size is found which satisfies the Armijo condition (3.6.2).

3. If no element of $T_k$ satisfies the Armijo condition, then set $t_k$ to be the adaptive step size.

For instance, in our numerical experiments reported in Section 3.9, we take $T_k$ to be $(1, \frac{1}{4}, \frac{1}{16})$ for all $k$. This allows the method to take steps of size $t_k = 1$ when 1 satisfies the Armijo condition, which is desirable for reducing the number of iterations needed before superlinear convergence kicks in.

We refer to this scheme as *hybrid step selection*. For a proper choice of $T_k$, hybrid step selection avoids the disadvantage of exclusively using adaptive step sizes, where the step size may be small for many iterations. It will also generally be more efficient to compute than a full line search, since no more than $|T_k|$ candidate step sizes are tested before switching to the adaptive step size.

### 3.8 Application to Stochastic Optimization

The adaptive step size can readily be extended to *stochastic* optimization methods. Consider a problem of the form

$$L(w) = \frac{1}{N} \sum_{i=1}^{N} f_i(w) + h(w). \tag{3.8.1}$$

If $N$ is extremely large, as is often the case in machine learning, simply evaluating $L(w)$ is an expensive operation, and line search is entirely impractical. To solve problems of the form (3.8.1), stochastic algorithms such as Stochastic Gradient Descent (SGD, [16]) select a random subset $S_k$ of $\{f_1, \ldots, f_N\}$ at step $k$ and compute the gradient for the subsampled problem

$$L^{(S_k)}(w) = \frac{1}{|S_k|} \sum_{f_i \in S_k} f_i(w) + h(w) \tag{3.8.2}$$

[1] As an illustrative example, consider applying the damped Newton method to the quadratic function $f(x) = \frac{1}{2}\|x\|^2$. Since $d_k = -x_k$ and $\delta_k = \|x_k\|$, we have $t_k = \frac{1}{1+\|x_k\|}$ and $x_{k+1} = \frac{\|x_k\|}{1+\|x_k\|} x_k$. If $\|x_0\|$ is large, then it is clear that the damped Newton method will take many tiny steps until $\|x_k\|$ is sufficiently reduced. This is in stark contrast to Newton's method, which reaches the minimizer after a single step.

as an approximation to the gradient of the loss function (3.8.1), and take a step using an empirically determined small and decreasing step size. In variance-reduced versions of SGD such as SVRG [20], it is common to use a constant step size, determined through experimentation. The curvature-adaptive step size has two desirable properties in this setting: it eliminates the need to select a step size through ad-hoc experimentation, and it incorporates second-order information, which is currently not exploited by most stochastic algorithms.

A detailed discussion of the theory of curvature-adaptive step sizes in the stochastic setting is beyond our current scope, as it depends heavily on the theory of empirical processes. A convergence analysis of stochastic gradient descent and stochastic BFGS with our adaptive step size can be found in [62].

There is currently also little work on algorithms exploiting the finite sum structure (3.8.1), which can provably attain superlinear convergence. Aside from [62], we are only aware of the Newton Incremental Method (NIM) of Rodomanov and Kropotov [63], and the DiSCO method of Zhang and Xiao [57], both of which are based on Newton's method. These methods require additional memory of the order $O(N)$, which is often substantial.

## 3.9 Numerical Experiments

### 3.9.1 Deterministic Methods

To compare our adaptive methods to classical algorithms, we solve several binary classification problems using *logistic regression*. In these problems, the objective function to be minimized has the form

$$L(w) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(-y_i x_i^T w)) + \frac{1}{2N}\|w\|_2^2. \tag{3.9.1}$$

where the training data $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ consists of feature vectors $x_i \in \mathbb{R}^n$ and classifications $y_i \in \{-1, +1\}$. Zhang and Xiao [57] showed that the logistic regression objective function $L(w)$ is self-concordant.

**Theorem 3.9.1** (Lemma 1, [57]). *Let $B = \max_i \|x_i\|$. The scaled function $\frac{B^2 N}{4} L(w)$ is standard*

*self-concordant.*

In our tests, we compared seven algorithms:

1. BFGS with adaptive step sizes (BFGS-A).
2. BFGS with Armijo-Wolfe line search (BFGS-LS).
3. BFGS with hybrid step selection (BFGS-H), using $T_k = (1, \frac{1}{4}, \frac{1}{16})$.
4. L-BFGS with adaptive step sizes (LBFGS-A), using the past $\ell = \min\{\frac{n}{2}, 20\}$ curvature pairs.
5. L-BFGS with Armijo-Wolfe line search (LBFGS-LS), using the past $\ell = \min\{\frac{n}{2}, 20\}$ curvature pairs.
6. Gradient descent with adaptive step sizes (GD-A).
7. Gradient descent with Armijo-Wolfe line search (GD-LS).

An initial Hessian approximation $H_0$ must be provided for the BFGS and L-BFGS methods. It is easy, but not necessarily effective, to simply take $H_0 = I$. Another common strategy for initializing $H_0$, described in [60], that is often quite effective, is to take $H_0 = I$ on the first step, and then, before performing the first BFGS update (3.6.1), scale $H_0$:

$$H_0 \leftarrow \frac{y_0^T s_0}{y_0^T y_0} I. \tag{3.9.2}$$

It is easy to verify that the scaling factor $y_0^T s_0 / y_0^T y_0$ lies between the smallest and largest eigenvalues of the inverse of the average Hessian $\overline{G} = \int_0^1 G(x_0 + \tau s_0) d\tau$ along the initial step.

Similarly, for the L-BFGS method, the initial matrix used at step $k+1$ in the two-loop recursion is chosen as:

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I.$$

We refer to this as *identity scaling*.

| Data set | $n$ | $N$ |
|---|---|---|
| covtype.libsvm.binary.scale | 55 | 581012 |
| ijcnn1.tr | 23 | 35000 |
| leu | 7130 | 38 |
| rcv1_train.binary | 47237 | 20242 |
| real-sim | 20959 | 72309 |
| w8a | 301 | 49749 |

Table 3.1: Data sets used in Section 3.9

The line search used the `WolfeLineSearch` routine from the minFunc software package [49]. The Armijo-Wolfe parameters were $c_1 = 0.1, c_2 = 0.75$, and the line search was configured to use an initial step size $t = 1$ and perform quadratic interpolation (`LS_interp = 1`, `LS_multi = 0`).

We chose six data sets from LIBSVM [52] with a variety of dimensions, which are listed in Table 3.1. We plot the progress of each algorithm as a function of CPU time used. The progress is measured by the *log gap* $\log_{10}(f(w) - f(w_*))$, where $w_*$ is a pre-computed optimal solution. The starting point was always set to $w_0 = 0$. All algorithms were terminated when either the gradient reached the threshold $\|g(x)\| < 10^{-7}$, or after 480 seconds of CPU time. A brief summary of the results can be found in Table 3.2, which lists the number of iterations taken by the BFGS-type methods for convergence.

Our algorithms were implemented in Matlab 2017a and run on an Intel i5-6200U processor. While the CPU time is clearly platform-dependent, we sought to minimize implementation differences between the algorithms to make the test results as comparable as possible.

In Figure 3.1, we plot the results for the data sets `covtype.libsvm.binary.scale`, `ijcnn1.tr`, and `w8a`. On these problems, we implemented BFGS with a dense Hessian; that is, the matrices $H_k$ were stored explicitly and updated using the formula (3.6.1). In Table 3.2, we list the number of iterations used by the BFGS-type methods.

In Figure 3.2, we plot the results for the data sets `leu`, `rcv1_train.binary`, and `real-sim`. These problems had a large number of variables ($n > 7000$), which made it infeasible to store $H_k$ explicitly. On these problems, BFGS was implemented using the two-loop recursion with unlim-

| Data set | $n$ | Identity Scaling | Number of iterations | | |
|---|---|---|---|---|---|
| | | | BFGS-A | BFGS-LS | BFGS-H |
| `covtype.libsvm.binary.scale` | 55 | No | 844 | 80 | 126 |
| | | Yes | 1532 | 458 | 479 |
| `ijcnn1.tr` | 23 | No | 286 | 36 | 66 |
| | | Yes | 434 | 142 | 162 |
| `w8a` | 301 | No | 2254 | 240 | 637 |
| | | Yes | 2506 | 398 | 653 |
| `leu` | 7130 | No | 1197 | 95 | 293 |
| | | Yes | 909 | 177 | 251 |
| `rcv1_train.binary` | 47237 | No | 161 | 31 | 35 |
| | | Yes | 284 | 217 | 232 |
| `real-sim` | 20959 | No | 356 | 44 | 55 |
| | | Yes | 592 | 247 | 317 |

Table 3.2: The number of iterations until convergence of the BFGS methods.

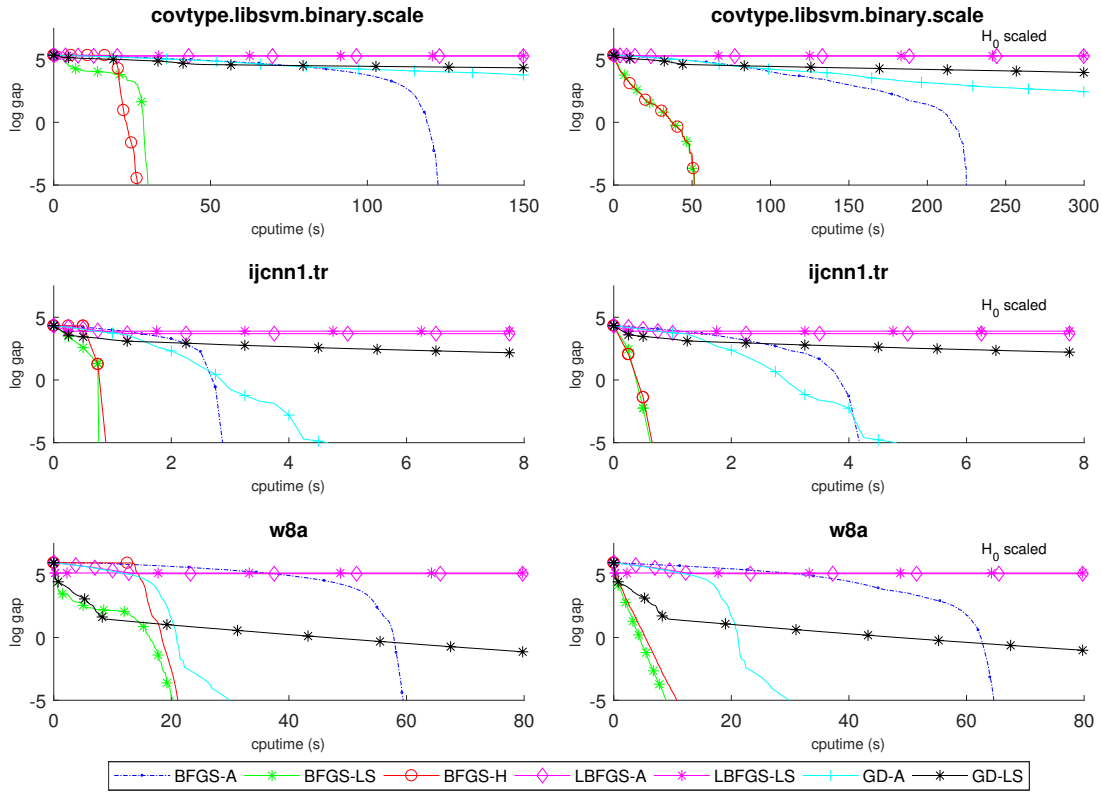


Figure 3.1: Experiments on problems with small $n$. The log gap is defined as $\log_{10}(f(w) - f(w_*))$. The loss functions are scaled to be standard self-concordant. All BFGS and L-BFGS plots on the left take $H_0 = I$, and those on the right use identity scaling.

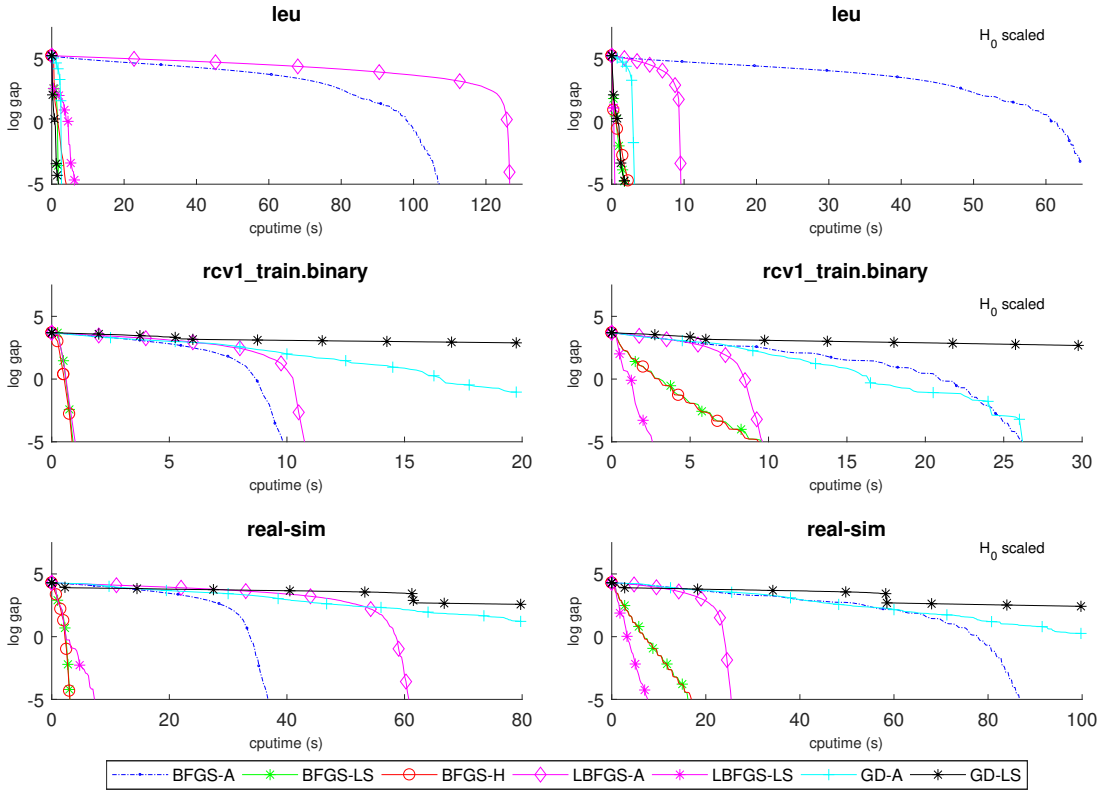


Figure 3.2: Experiments on problems with large $n$. The log gap is defined as $\log_{10}(f(w) - f(w_*))$. The loss functions are scaled to be standard self-concordant. All BFGS and L-BFGS plots on the left take $H_0 = I$, and those on the right use identity scaling.

ited memory, and $H_0$ was kept fixed throughout the iteration process. If the number of iterations exceeds roughly $n/4$, then this approach would in fact require more memory than storing $H_k$ explicitly. However, this never occurred in our tests, as shown in Table 3.2.

In our tests, we found that BFGS-A required more time than BFGS-LS. Although the cost of a single step was initially lower for BFGS-A than BFGS-LS, BFGS-A often took numerous small steps in succession, making very slow progress. This situation was exactly our motivation for devising the hybrid step selection described in Section 3.7, and unfortunately, appears to occur often. However, BFGS-H achieved comparable speed to that of BFGS-LS with $T = (1, \frac{1}{4}, \frac{1}{16})$, which suggests that always trying $t = 1$ first is an excellent heuristic. These results also provide evidence of the effectiveness of performing inexact line searches, in settings where it is practical to do so. In Table 3.3, the number of steps needed until we consistently have $t_k \approx 1$ is shown.

Since computing $t_k$ also requires a Hessian-vector product, the cost comparison between the

| Data set | $n$ | Identity Scaling | Number of iterations | | |
|---|---|---|---|---|---|
| | | | BFGS-A | BFGS-LS | BFGS-H |
| `covtype.libsvm.binary.scale` | 55 | No | 797 | 57 | 62 |
| | | Yes | 1378 | 2 | 2 |
| `ijcnn1.tr` | 23 | No | 270 | 25 | 26 |
| | | Yes | 369 | 3 | 2 |
| `w8a` | 301 | No | 2056 | - | 289 |
| | | Yes | 2250 | 5 | 2 |
| `leu` | 7130 | No | - | - | 42 |
| | | Yes | 818 | 2 | 2 |
| `rcv1_train.binary` | 47237 | No | 132 | 15 | 4 |
| | | Yes | 205 | 3 | 4 |
| `real-sim` | 20959 | No | 294 | 18 | 17 |
| | | Yes | 490 | 2 | 2 |

Table 3.3: The number of iterations until $t_k = 1$ was consistently accepted by BFGS-LS and BFGS-H, and, for BFGS-A, the number of iterations until $t_k \geq 0.9$ for at least 80% of the remaining iterations. A dash '-' indicates that the condition was not met before the stopping criterion was satisfied.

adaptive step size and inexact line search reverses when the algorithm nears convergence. Initially, a Hessian-vector product is faster than performing multiple backtracking iterations and repeatedly testing for the Armijo-Wolfe conditions; however, the line search (and the hybrid step selection) will eventually accept the step size $t_k = 1$ immediately, becoming essentially free, whereas computing the adaptive step size continues to require a Hessian-vector product on every step.

Curiously, L-BFGS was far more effective on the problems with large $n$ (Figure 3.2) than on those with small $n$ (Figure 3.1). Both LBFGS-A and LBFGS-LS were ineffective on the problems with small $n$, which suggests that the problem lies with the step directions computed by L-BFGS, rather than the step sizes. Identity scaling was also beneficial for L-BFGS on problems with large $n$, substantially reducing the convergence time in some cases. We note that we did not experiment comprehensively with varying $\ell$, the number of curvature pairs stored in L-BFGS, and used a standard choice of $\ell = \min\{\frac{n}{2}, 20\}$. Other choices of $\ell$ might lead to very different results on the problems in our test set.

On the other hand, identity scaling appeared to be detrimental for the BFGS-type methods on most problems, which can be seen from the plots in Figure 3.1 and Figure 3.2 by comparing the

CPU time needed for convergence. For instance, on the data set `covtype.libsvm.binary.scale`, the time to convergence for BFGS-A increased from 120s to 225s, and from 25s to 50s for BFGS-LS and BFGS-H. In fact, identity scaling was beneficial for the BFGS-A method *only* on the data set `leu`. The data set `leu` appears to be quite different from the other problems tested. The number of training samples for `leu` was $m = 38$, while for all other problems, $m$ was at least 20,000. Moreover, gradient descent with Armijo-Wolfe line search (GD-LS) was among the fastest methods on `leu`, while on the other test problems it was significantly outperformed by BFGS. The iteration counts shown in Table 3.2 and Table 3.3 also indicate that identity scaling worsened the performance of the BFGS methods on every problem except `leu`. Curiously, performing identity scaling led to BFGS-H accepting $t_k = 1$ at a much earlier iteration on all problems, yet the total CPU time used by BFGS-H was longer for `covtype.libsvm.binary.scale`, `rcv1.train.binary`, and `real-sim`.

GD-A was surprisingly effective, outperforming GD-LS on every problem except for `leu`. This is somewhat surprising (in light of the performance of BFGS-A and BFGS-LS), and suggests that the curvature-adaptive step size may be useful for selecting hyperparameters for first-order methods.

### 3.9.2 Stochastic Methods

The experiments presented here are derived from the experiments in [62, §4]. Several stochastic algorithms are tested on an *online least-squares* problem of the form

$$\min_w \mathbb{E}(Y - X^T w)^2 + \frac{1}{2}\lambda\|w\|^2.$$

*Online* refers to the method of sampling: we can only access $(X, Y)$ by calling an oracle at each iteration $k$, which returns $|S_k|$ i.i.d instances of $(X, Y)$. The model for $(X, Y)$ has the following specification:

- $X$ has a multivariate normal distribution $N(0, \Sigma)$, where $\Sigma$ is the covariance matrix of the

w8a data set (see Table 3.1).

- $Y = X^T\beta + \epsilon$, where $\beta$ is deterministic and sparse (80% sparsity) and $\epsilon \sim N(0,1)$ is a noise component.

Since our model is based on the w8a data set, the dimension of $w$ is $p = 300$, and the regularizer is set to $\lambda = \frac{1}{p}$.

We compare the following algorithms. For a deterministic method $M$, the corresponding *stochastic $M$ method* takes the step of the underlying $M$ method, but computed from the empirical objective function (3.8.2) sampled at each iteration. The convergence of these methods[2] is analyzed in [62]. In summary, the stochastic adaptive gradient descent method returns an $\epsilon$-optimal solution in expectation after $O(\log(\epsilon^{-1}))$ iterations when $|S_k|$ is chosen as a constant (depending on $\epsilon$), and stochastic adaptive BFGS converges $R$-superlinearly with probability 1 when $|S_k|$ increases superlinearly.

**SBFGS-A** The stochastic adaptive BFGS method. At each iteration, an adaptive BFGS step is computed from the empirical objective function (3.8.2). The BFGS update is computed from the pair $(d_k, G_k d_k)$ which is more stable than using the pair $(s_k, y_k)$ with the difference $y_k$ of sampled gradients (see [32, 31]).

**SBFGS-1** The stochastic BFGS method with *constant* step size $\alpha_1$. The step size $\alpha_1$ is given in Table 3.4.

**SN-A** Nesterov's stochastic damped Newton method [33].

**SN-1** The stochastic Newton method with constant step size $\alpha_1$.

**SGD-A** Stochastic adaptive gradient descent.

**SGD-$i$** Stochastic gradient descent with constant step size $\alpha_i$ for $i = 1, \ldots, 4$ (Table 3.4).

[2]Note: the stochastic adaptive BFGS analyzed in [62] is slightly different, as it incorporates an additional Wolfe condition test.

| | Value |
|---|---|
| $\alpha_1$ | $\frac{1}{140{,}000} \approx$ `7.14e-6` |
| $\alpha_2$ | `5e-6` |
| $\alpha_3$ | `2e-6` |
| $\alpha_4$ | `1e-6` |

Table 3.4: Constant step sizes.

The theory [62] suggests taking an increasing number of samples for stochastic adaptive methods. For SBFGS-A, SBFGS-1, SN-A, SN-1, and SGD-A, we use $|S_k| = \frac{1}{2}p \cdot (1.05)^{\lfloor \frac{k}{50} \rfloor}$. That is, the number of samples starts at $\frac{1}{2}p = 150$ and increases by 5% every 50 iterations. For SGD-$i$ methods, we test three different constant batch sizes: a *small* batch of $|S_k| = \frac{1}{2}p$, a *medium* batch of $|S_k| = p$, and a *large* batch of $|S_k| = 4p$.

The results of the experiments are shown in Figure 3.3. As before, the *log gap* is $\log_{10}(f(x_k) - f(x_*))$, where $x_*$ is the true minimizer ($x_*$ can be computed explicitly given $\Sigma$ and $\beta$). The plots in the first column shows the trajectory of each method in 60 seconds of CPU time; the second and third columns show the final 10 seconds (from 50s to 60s) in greater detail. The starting point in all trials was $w = 0$.

Both SBFGS-A and SN-A exhibit superlinear convergence once they approach the minimizer. Curiously, SBFGS-A attains greater accuracy than SN-A using the same sample sizes (see second column of Figure 3.3); we suspect that the noisiness of sampling $G_k$ damages SN-A. These methods greatly outperform SGD, even with well-tuned step sizes. We note that SGD is quite sensitive to the choice of step size. A constant step size cannot be made much larger than $\alpha_1$; using $\frac{1}{130{,}000} \approx$ `7.69e-6` causes SGD (even with large batches) to immediately diverge. In fact, we can check that the largest eigenvalue of $\Sigma$ is approximately `1.32e5`. Furthermore, the superior performance of SBFGS-A and SN-A depends at least partially on the curvature-adaptive step size. The methods SBFGS-1 and SN-1, which use the constant step size $\alpha_1$, converge extremely slowly[3], so the success of SBFGS-A and SN-A is not solely due to the second-order information in $H_k$.

[3]It is possible to use even larger step sizes with these methods. We observed that stochastic BFGS and stochastic Newton can tolerate much larger constant step sizes than $\alpha_1$ without diverging wildly as SGD does. However, for stochastic BFGS, the performance is not better, and is usually much worse than using $\alpha_1$, as the algorithm escapes to a worse region before beginning to decrease slowly.

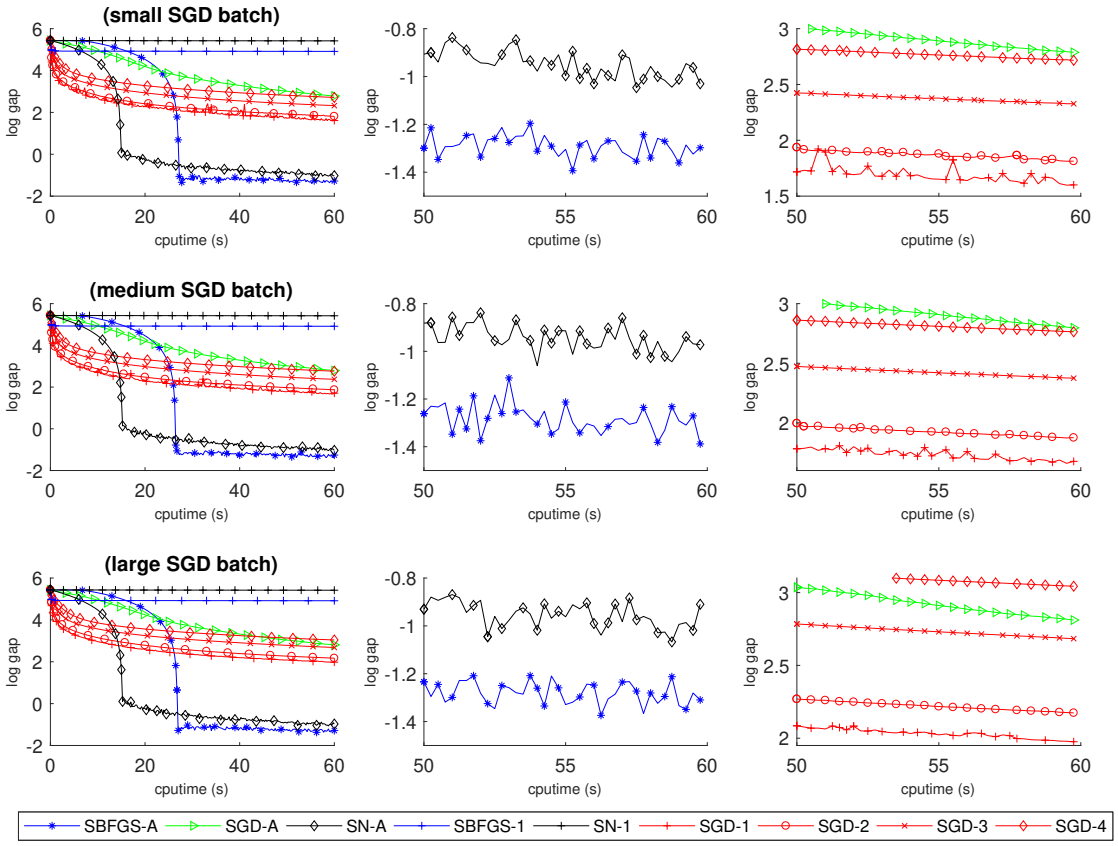


Figure 3.3: Performance of the stochastic algorithms. In the top row, the SGD methods SGD-1, SGD-2, SGD-3, SGD-4 use small batches ($|S_k| = \frac{1}{2}p$). Likewise, the second and third row use medium and large batches, respectively. The first column shows the performance of each method in 60s of CPU time, and the second and third columns show a close-up of the last 10s (50s-60s).

SGD-A was slower than SGD with tuned step sizes. We found that the initial adaptive step size was `1e-8`, which explains the relatively slow convergence of SGD-A. It is also worth noting that even with a small initial sample, SGD-A never produced an overly large step size causing it to diverge or oscillate, something which is not strictly guaranteed by the theory.

We have not touched on the subject of variance reduction, which is generally crucial, though not particularly relevant when considering the results in Figure 3.3. Good variance reduction techniques will be important for designing an effective, general-purpose solver based on SBFGS-A, SN-A, or indeed, any other of the stochastic algorithms tested.

# Chapter 4: Distributed Optimization: The Leader Stochastic Gradient Descent Algorithm

## 4.1 Introduction

The advent of increasingly large data and models has led to the use of *parallel* and *distributed* computing for machine learning [36, 64]. A number of algorithms [65, 38, 66, 67] and systems (DOWNPOUR, Horovod) [68, 69] have been proposed for specifically taking advantage of parallelism. Consider again the loss function encountered in a typical supervised learning problem:

$$\min_x f(x) = \frac{1}{N}\sum_{i=1}^{N} \ell(x;\xi_i)$$

where both $N$, the size of the training data $\{\xi_1, \ldots, \xi_N\}$, as well as the dimension $d$ of the model parameters $x$, may be large. A straightforward way to parallelize the standard Stochastic Gradient Descent (SGD) algorithm is *data parallelism*: given $p$ parallel workers, the minibatch $B$ at each step is partitioned into $p$ sets $B_1, \ldots, B_p$, and each worker computes the gradient $g_j = \sum_{\xi_i \in B_j} \nabla \ell(x;\xi_i)$. A central coordinator then aggregates the gradients and computes the stochastic gradient $\sum_{j=1}^{p} g_j$ for SGD, performing an update and communicating new parameters $x^+$ to the workers.

The data-parallel strategy is simple and easy to implement. However, it has two notable drawbacks. The first is that it quickly encounters *limits to parallelism*. The limiting factor is the *size of the minibatch*: a well-known empirical phenomenon is that *generalization* is often poor when using SGD with large minibatches [70, 71]. This is not entirely understood, and is often attributed to the *flat minima theory*, which hypothesizes that local minima belonging to 'flatter' regions of the loss landscape result in better generalization. Algorithms have been proposed to modify SGD to converge to such 'flat' minima [72, 73]. Another common strategy is to increase the step size

when using large minibatches [74, 37].

The second drawback of data parallelism is the *frequency of communication*. At every step, each worker must communicate the gradient of its assigned subset to the central coordinator, and the coordinator then must communicate updated parameters to every worker. Though these operations have efficient implementations and are specifically supported by computing frameworks such as MPI (via the *reduce* and *scatter* operations), requiring synchronization between all workers and the coordinator can lead to significant and unpredictable communication delays in practice. Allowing less strict synchronization leads to 'asynchronous' algorithms which have convergence guarantees under various assumptions on the delay [75, 76], it is still often unclear whether such delay models are realistic or whether good performance is obtained in practice.

In a data-parallel algorithm, there is a single set of model parameters shared by all the workers at each step. Another paradigm for distributed optimization allows for multiple independent instances of the parameters being separately optimized. Unlike data-parallel algorithms, these algorithms can allow for *reduced communication*, because we do not require that the separate instances of the model share weights at every point in time. A key algorithm of this type is *Elastic Averaging SGD* [38, 77], which uses an $L_2$-penalty to enforce a 'soft' consensus between the models on separate workers. To specify this algorithm, assume that we have $p$ independent workers, each having its own copy $x^{(i)}$ of the parameters, and let $f(\cdot)$ denote the common objective function. In the machine learning context, note that $f$ denotes the total loss $f(x) = \frac{1}{N}\sum_{i=1}^{N} \ell(x; \xi_i)$ – we assume each worker to have access to the same training dataset $\{\xi_1, \ldots, \xi_N\}$. Finally, let $\widetilde{x}$ be another decision variable, which we call the consensus variable. Each worker performs the update

$$x_{k+1}^{(i)} = x_k^{(i)} - \eta(\nabla f(x_k^{(i)}) + \lambda(x_k^{(i)} - \widetilde{x}_k)).$$

It is not hard to see that this corresponds to performing a gradient descent step with respect to $x^{(i)}$ for the loss function

$$f(x) + \frac{\lambda}{2}\|x - \widetilde{x}\|^2.$$

In the stochastic case (for *SGD* instead of GD), the gradient $\nabla f(x^{(i)})$ is replaced by a stochastic gradient estimate. In parallel to the workers updating their parameters $x_k^{(i)}$, the consensus variable in EASGD is updated by a moving average of the model parameters:

$$\widetilde{x}_{k+1} = (1-\beta)\widetilde{x}_k + \beta\left(\frac{1}{p}\sum_{i=1}^{p} x_k^{(i)}\right)$$

In practice, we can reduce communication by performing *multiple* gradient steps for each set of parameters $x^{(i)}$, and either holding the consensus variable $\widetilde{x}_k$ fixed to its value at step $k$, or including the consensus term $\lambda(x_k^{(i)} - \widetilde{x}_k)$ only for the very first step. Since each worker can perform SGD independently for $f(x^{(i)})$, the only communication is a periodic *gather* operation to average the worker parameters. This variant is called the *Asynchronous EASGD*.

EASGD has been successfully applied to deep learning [38], and is widely used in industry[1]. It is interesting that EASGD has proven to be effective for deep learning, a problem domain where the loss function is typically *highly nonconvex*, while the known convergence analysis only applies to convex problems. In fact, we can show that for nonconvex problems, EASGD has stationary points which are "spurious" in the sense of not corresponding to stationary points of the underlying objective function $f$.

**Proposition 4.1.1.** *Let $p = 2$. There exists a Lipschitz differentiable function $f : \mathbb{R} \to \mathbb{R}$ such that for every $0 < \lambda \leq 1$, there exists a point $(x_\lambda, y_\lambda, 0)$ which is a stationary point of EASGD with parameter $\lambda$, but none of $\{x_\lambda, y_\lambda, 0\}$ is a stationary point of $f$.*

We prove this proposition in the next section.

To see how this can theoretical pitfall can impact convergence, observe that on nonconvex problems, the workers $x^{(i)}$ may converge towards different local minima, causing the consensus variable (which converges to the average of the worker parameters) to pull the workers in the wrong direction. In particular, when the objective landscape is symmetric, the consensus variable can become permanently trapped at a local maximum in between the local minima. Symmetry

[1] Personal communication.

is a common feature in problems involving representation learning [78, 79, 80, 81] and in deep learning [82, 83].

We propose a simple modification of EASGD that eliminates this pitfall (see Proposition 4.8.2). Consider the global loss function (summed over workers) with the consensus term:

$$\min_x \mathcal{L}(x^{(1)}, \ldots, x^{(p)}, \widetilde{x}) = \frac{1}{p}\sum_{i=1}^{p} f(x^{(i)}) + \frac{\lambda}{2}\|x^{(i)} - \widetilde{x}\|^2$$

In EASGD, the consensus variable $\widetilde{x}$ is a decision variable which is updated using a moving average.

Instead, we set

$$\widetilde{x} = \operatorname{argmin}\{f(x^{(1)}), f(x^{(2)}), \ldots, f(x^{(p)})\}$$

so that $\widetilde{x}$ induces a pulling towards the parameters which had the best objective value. The resulting gradient step then takes the form

$$x_{k+1}^{(i)} = x_k^{(i)} - \eta(\nabla f(x_k^{(i)}) + \lambda(x_k^{(i)} - \widetilde{x}_k)) = (1 - \eta\lambda)x_k^{(i)} + \eta\lambda\widetilde{x}_k - \eta\nabla f(x_k^{(i)})$$

which is identical to that of EASGD except for the definition of $\widetilde{x}_k$. We call this method *Leader (Stochastic) Gradient Descent* (L(S)GD), and refer to the consensus variable $\widetilde{x}$ defined using $\operatorname{argmin}$ as the *leader* or *guiding* point.

To make this method into a *practical* algorithm for large-scale problems, there are several relaxations which must be made. First, rather than using the exact gradient $\nabla f(x^{(i)})$, we typically prefer a *stochastic* gradient descent, so we instead have an estimator $\widetilde{\nabla} f(x^{(i)})$. The leader variable $\widetilde{x}$ may not be exact, but can contain error from two sources:

- To reduce communication, the leader $\widetilde{x}$ may be held fixed for multiple steps, so it will not necessarily satisfy $f(\widetilde{x}) \leq f(x_k^{(i)})$ at every time $k$. The delay in communication might be an explicit choice in the algorithm, or can be caused by unpredictable slowness in the workers or network.

- We may use noisy estimates $\widetilde{f}(x^{(1)}), \ldots, \widetilde{f}(x^{(p)})$ of the function values, so $\widetilde{x}$ will be set to a worker's parameters other than the true minimizer with some probability.

In our theoretical analysis, we address these scenarios. We first consider an 'ideal' LSGD where the leader is guaranteed to satisfy $f(\widetilde{x}) \leq f(x_k^{(i)})$ at every step. We obtain the convergence rate of this algorithm in the standard setting for SGD, namely strongly convex stochastic optimization (Section 4.5). We then proceed to analyze the effects of relaxing the leader point: allowing for delays in communication (Section 4.6) and stochastic leader selection (Section 4.7). We then turn to consider nonconvex optimization, and show that the deterministic LGD has strong guarantees (Section 4.8).

We also investigate other properties of L(S)GD. In Section 4.9, we define a notion of *improving the search direction*, and for the positive definite quadratic model, show that a large subset of the possible leader points result in improvement. In Section 4.10, we give a negative result: unlike EASGD, LSGD is not able to implicitly reduce the limiting variance by increasing the number of workers.

### 4.2 Motivating Example: Matrix Factorization

Before proceeding to the theoretical analysis, we give a practical demonstration of how symmetry and Proposition 4.1.1 can result in poor convergence of EASGD. Consider the *low-rank matrix factorization* problem, which is a nonconvex learning problem whose landscape exhibits numerous symmetries. It is known that every local minimum is global [67]. We consider the positive semidefinite case, where the objective is to find a low-rank matrix minimizing

$$\min_X \left\{ f(X) = \frac{1}{4}\|M - XX^T\|_F^2 : X \in \mathbb{R}^{d\times r} \right\}.$$

We compare the (deterministic) EAGD and LGD algorithms. It is routine to calculate that

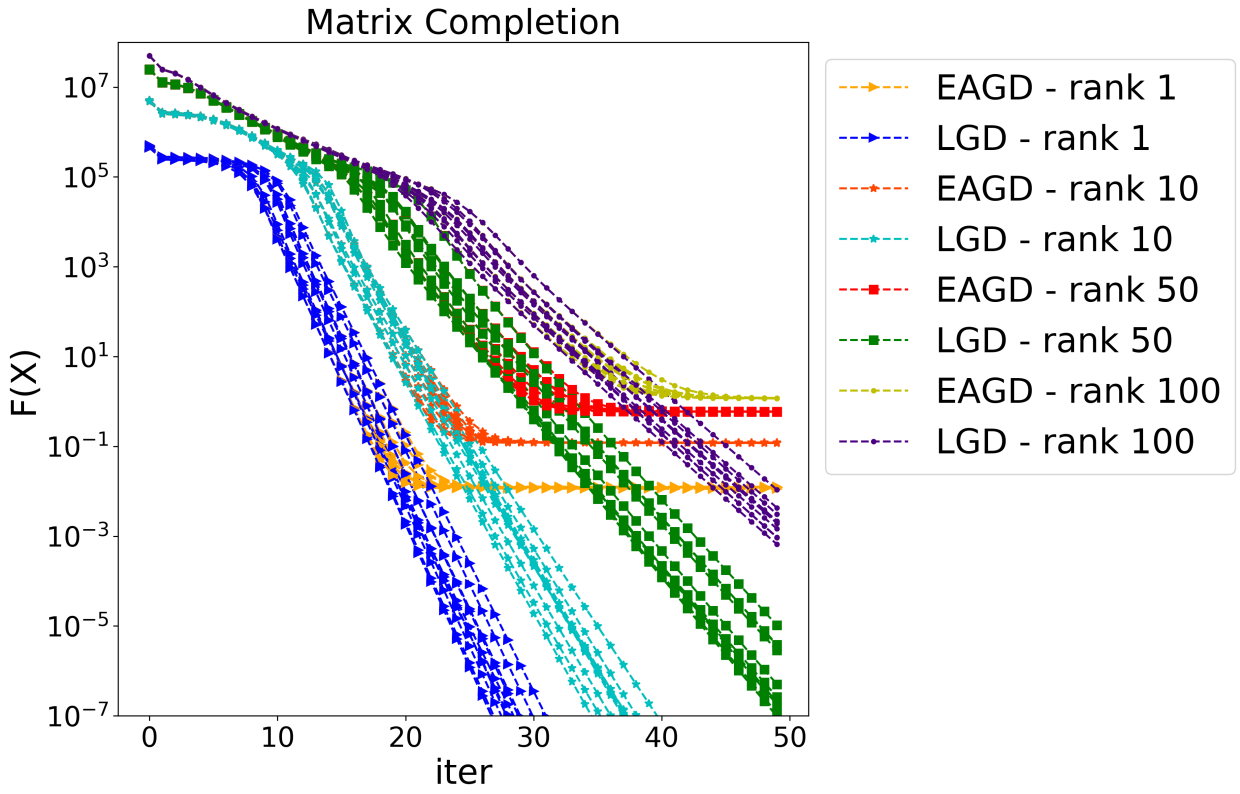


Figure 4.1: Low-rank matrix factorization problems solved with EAGD and LGD. The dimension $d = 1000$ and four ranks $r \in \{1, 10, 50, 100\}$ are used. The reported value for each algorithm is the value of the best worker (8 workers are used in total) at each step.

$\nabla F(X) = (XX^T - M)X$. The EASGD and LSGD updates for $X$ can be expressed as

$$X_+ = (1 - \eta\lambda)X + \eta\lambda Z - \eta\nabla F(X).$$

For EASGD, $Z = \widetilde{X}$, and $\widetilde{X}$ is updated by

$$\widetilde{X}_+ = (1 - p\eta\lambda)\widetilde{X} + p\eta\lambda \left(\frac{1}{p}\sum_{i=1}^{p} X^i\right).$$

For LSGD, $Z = \operatorname{argmin}\{F(X^1), \ldots, F(X^p)\}$, and is updated at the beginning of every step.

For four choices of the rank $r$, we generated $10$ random instances of the matrix completion problem, and solved each with EAGD and LGD, initialized from the same starting points (we use $8$ workers). For each algorithm, we report the progress of the *best* objective value at each iteration, over all workers. Figure 4.1 shows the results across $10$ random experiments for each rank.

It is clear that EAGD slows down significantly as it approaches a minimizer. Typically, the center $\widetilde{X}$ of EAGD is close to the average of the workers, which is a poor solution for the matrix completion problem when the workers are approaching different local minimizers, even though all local minimizers are globally optimal. This induces a pull on each node *away* from the minimizers, which makes it extremely difficult for EAGD to attain a solution of high accuracy. In comparison, LGD does not have this issue.

### 4.3 Definitions and Preliminaries

Throughout our proofs in this section, we will often consider a more general form of the algorithm where the consensus variable is replaced by an arbitrary point $z$. That is, we optimize

$$\mathcal{L}(x^{(1)}, \ldots, x^{(p)}) = \frac{1}{p}\sum_{i=1}^{p} f(x^{(i)}) + \frac{\lambda}{2}\|x^{(i)} - z\|^2 \tag{4.3.1}$$

where $z$ may be given by weaker properties.

A L(S)GD step (with respect to a given $z$) is a (stochastic) gradient step applied to $\mathcal{L}$: writing

$z = \widetilde{x}$ at a particular $(x^{(1)}, \ldots, x^{(p)})$, the step in the variable $x^{(i)}$ is

$$\eta(\widetilde{\nabla} f(x^{(i)}) + \lambda(x^{(i)} - z)).$$

The deterministic LGD algorithm uses the exact gradient $\widetilde{\nabla} f(x^{(i)}) = \nabla f(x^{(i)})$, and the stochastic LSGD algorithm samples an unbiased estimator with $\mathbb{E}[\widetilde{\nabla} f(x^{(i)})] = \nabla f(x^{(i)})$. This estimator is assumed to satisfy standard growth conditions on its variance, which are given in the analysis.

Observe that if $z$ is defined as the standard LSGD point $\operatorname{argmin}\{f(x^{(1)}, \ldots, x^{(p)}\}$, then for the index $i$ such that $z = x^{(i)}$, the step for $x^{(i)}$ reduces to a gradient step.

To reduce communication costs in the distributed setting, we may choose to infrequently update the leader. For each time $k$, the variable $x^{(i)}$ will take $b_i(k)$ L(S)GD steps, using the same leader fixed from the beginning (alternately, we compare the leader with the function value of $x^{(i)}$ only). This number $b_i(k)$ is the *communication period*. Typically, $b_i(k)$ is constant for all workers $i$ and all steps $k$, and we simply write $\tau$ for the communication period.

After each variable takes its $b_i(k)$ steps, the function values are computed and the next leader is detremined. Let $\widetilde{x}^{(k)}$ denote the leader for the $k$-th period, and let $x_{k,j}^{(i)}$ denote the value of the variable $x^{(i)}$ after taking $j$ steps in the period $k$.

In practice, we may use multiple leaders to increase locality and reduce communication costs, such as in the group method described in [3]. This arises because of natural clustering in the hardware, and heterogeneity in communication costs. For example, a standard setup consists of multiple machines for which communication is expensive, but with each machine having multiple GPUs internally for which data transfer is fast between GPUs. For our theory, it suffices to consider having only one leader. The formulation with multiple leaders (for one variable) is given by

$$\min_x f(x) + \frac{\lambda_1}{2}\|x - z_1\|^2 + \ldots + \frac{\lambda_c}{2}\|x - z_c\|^2$$

However, this is equivalent to minimizing $f(x) + \frac{\Lambda}{2}\|x - \widetilde{z}\|^2$, where $\Lambda = \sum_{i=1}^c \lambda_i$ and $\widetilde{z} = \frac{1}{\Lambda}\sum_{i=1}^c \lambda_i z_i$. Thus, for our theoretical analysis, we may reduce to the case of a single leader.

### 4.4 Stationary Points of EASGD

We first prove Proposition 4.1.1, which shows that EASGD can converge to spurious stationary points when the objective function is nonconvex.

**Proposition.** *Let $p = 2$. There exists a Lipschitz differentiable function $f : \mathbb{R} \to \mathbb{R}$ such that for every $0 < \lambda \leq 1$, there exists a point $(x_\lambda, y_\lambda, 0)$ which is a stationary point of EASGD with parameter $\lambda$, but none of $\{x_\lambda, y_\lambda, 0\}$ is a stationary point of $f$.*

*Proof.* Define $f(x)$ by

$$f(x) = \begin{cases} e^{x+1} & \text{if } x < -1 \\ q(x) & \text{if } -1 \leq x \leq 1 \\ e^{-x+1} & \text{if } x > 1 \end{cases}$$

where $q(x) = a_6x^6 + \ldots + a_1x + a_0$ is a sixth-degree polynomial. For $f$ to be Lipschitz differentiable, we will select $q(x)$ to make $f$ twice continuously differentiable, with bounded second derivative. To make $f$ twice continuously differentiable, we must have $q(1) = 1, q'(1) = -1, q''(1) = 1$ and $q(-1) = -1, q'(-1) = 1, q''(-1) = -1$. Since we aim to have $f'(0) \neq 0$, we also will require $f'(0) = q'(0) = 1$. The existence of $q$ is equivalent to the solvability of the linear system

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 6 & -5 & 4 & -3 & 2 & -1 & 0 \\ 30 & 20 & 12 & 6 & 2 & 0 & 0 \\ 30 & -20 & 12 & -6 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

which is easily seen to be solvable. Thus, we deduce that such a function $f$ exists.

Suppose that we have two workers, and write $x = x^{(1)}, y = x^{(2)}$. Assume also that we have exact gradients (so the problem is deterministic). It remains to show that for any $0 < \lambda \leq 1$,

there exists a stationary point $(x, y, 0)$ of EASGD. When the consensus variable satisfies $\widetilde{x} = 0$, the first-order condition for $x$ yields $f'(x) + \lambda x = 0$. Since $\lambda \leq 1$, we have $\lambda(1) + f'(1) \leq 0$. For $x \geq 1$, $f'(x) = -e^{-x+1}$ is an increasing function, so $f'(x) + \lambda x$ is increasing, and we deduce that there exists a solution $y_\lambda \geq 1$ with $\lambda y_\lambda + f'(y_\lambda) = 0$. By symmetry, $-y_\lambda \leq -1$ satisfies $f'(-y_\lambda) + \lambda(-y_\lambda) = 0$, since $f'(x) = e^{x+1}$ for $x \leq -1$.

Since $x_\lambda = -y_\lambda$, the EASGD update of the consensus variable yields

$$\begin{aligned}\widetilde{x}_+ &= (1-\beta)\widetilde{x} + \frac{\beta}{2}(x_\lambda + y_\lambda) \\ &= (1-\beta)\widetilde{x} = 0\end{aligned}$$

Hence, $(-y_\lambda, y_\lambda, 0)$ is a stationary point of EASGD, but none of $\{-y_\lambda, y_\lambda, 0\}$ are stationary points of $f$. □

### 4.5 Convergence Rates for Stochastic Convex Optimization

Our key technical result is that LSGD satisfies a similar one-step descent in expectation as SGD, with an additional term corresponding to the pull of the leader. To provide a unified analysis of 'pure' LSGD as well as more practical variants where the leader is updated infrequently or with errors, we consider a general iteration $x_+ = x - \eta(\widetilde{g}(x) + \lambda(x - z))$, where $z$ is an arbitrary guiding point; that is, $z$ may not be the minimizer of $x^{(1)}, \ldots, x^{(p)}$, nor even satisfy $f(z) \leq f(x^{(i)})$. Since the nodes operate independently except when updating $z$, we may analyze LSGD steps for each node individually, and we write $x = x^{(i)}$ for brevity.

**Assumption 1** $f$ is $M$-Lipschitz-differentiable and $m$-strongly convex, which is to say, the gradient $\nabla f$ satisfies $\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|$, and $f$ satisfies

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}\|y - x\|^2.$$

We write $x^*$ for the unique minimizer of $f$, and $\kappa := \frac{M}{m}$ for the condition number of $f$.

**Lemma 4.5.1** (One-Step Descent)**.** *Let $f$ satisfy Assumption 1, and let $\widetilde{g}(x)$ be an unbiased estimator for $\nabla f(x)$ with variance $\mathrm{Var}(\widetilde{g}(x)) \leq \sigma^2 + \nu\|\nabla f(x)\|^2$.*

*Fix an initial point $x$, and let $z$ be another point, with $\delta := x - z$. We take a LSGD step*

$$x_+ = x - \eta(\widetilde{g}(x) + \lambda(x - z)).$$

*Then each step of LSGD satisfies:*

$$\begin{aligned}\mathbb{E}f(x_+) \leq f(x) &- \frac{\eta}{2}(1 - \eta M(\nu + 1))\|\nabla f(x)\|^2 \\ &- \frac{\eta}{4}\lambda(m - 2\eta M\lambda)\|\delta\|^2 \\ &- \frac{\eta\sqrt{\lambda}}{\sqrt{2}}(\sqrt{m} - \eta M\sqrt{2\lambda})\|\nabla f(x)\|\|\delta\| \\ &- \eta\lambda(f(x) - f(z)) + \frac{\eta^2}{2}M\sigma^2\end{aligned} \tag{4.5.1}$$

*Hence, for sufficiently small $\eta, \lambda$ with $\eta \leq (2M(\nu + 1))^{-1}$ and $\eta\lambda \leq (2\kappa)^{-1}, \eta\sqrt{\lambda} \leq (\kappa\sqrt{2m})^{-1}$, we have*

$$\mathbb{E}f(x^+) - f(x^*) \leq (1 - m\eta)(f(x) - f(x^*)) - \eta\lambda(f(x) - f(z)) + \frac{\eta^2 M}{2}\sigma^2 \tag{4.5.2}$$

*Proof.* The proof is similar to the convergence analysis of SGD. We take a Taylor expansion at the point $x$:

$$f(x_+) = f(x) - \eta\nabla f(x)^T(\widetilde{g}(x) + \lambda\delta) + \frac{\eta^2}{2}(\widetilde{g}(x) + \lambda\delta)^T G(\widetilde{g}(x) + \lambda\delta)$$

where $G = \nabla^2 f(x')$ for some $x'$ between $x, x_+$. Taking the expectation and using $\mathbb{E}\widetilde{g}(x) = \nabla f(x)$,

$$\mathbb{E}f(x_+) = f(x) - \eta\|\nabla f(x)\|^2 - \eta\lambda\nabla f(x)^T\delta + \frac{\eta^2\lambda^2}{2}\delta^T G\delta + \eta^2\lambda\nabla f(x)^T G\delta + \frac{\eta^2}{2}\mathbb{E}[\widetilde{g}(x)^T G\widetilde{g}(x)]$$

Taking another Taylor expansion along the direction $-\delta$, observe that

$$f(z) = f(x) - \nabla f(x)^T \delta + \frac{1}{2}\delta^T \widetilde{G}\delta \geq f(x) - \nabla f(x)^T \delta + \frac{m}{2}\|\delta\|^2$$

from which we deduce that $-\nabla f(x)^T \delta \leq -(f(x) - f(z) + \frac{m}{2}\|\delta\|^2)$. Substituting this above, and splitting both the terms $\eta\|\nabla f(x)\|^2, \frac{\eta}{2}m\lambda\|\delta\|^2$ in half, we obtain

$$\begin{aligned}
\mathbb{E}f(x_+) = f(x) &- \frac{\eta}{2}\|\nabla f(x)\|^2 + \frac{\eta^2}{2}\mathbb{E}[\widetilde{g}(x)^T G\widetilde{g}(x)] \\
&- \frac{\eta}{4}m\lambda\|\delta\|^2 + \frac{\eta^2}{2}\lambda^2\delta^T G\delta \\
&- \frac{\eta}{2}\|\nabla f(x)\|^2 - \frac{\eta}{4}m\lambda\|\delta\|^2 + \eta^2\lambda\nabla f(x)^T G\delta \\
&- \eta\lambda(f(x) - f(z))
\end{aligned}$$

We proceed to bound each line. For the first line, the standard bias-variance decomposition yields

$$\mathbb{E}[\widetilde{g}(x)^T G\widetilde{g}(x)] \leq M\mathbb{E}\|\widetilde{g}(x)\|^2 \leq M((\nu+1)\|\nabla f(x)\|^2 + \sigma^2)$$

and so

$$-\frac{\eta}{2}\|\nabla f(x)\|^2 + \frac{\eta^2}{2}\mathbb{E}[\widetilde{g}(x)^T G\widetilde{g}(x)] \leq -\frac{\eta}{2}(1 - \eta M(\nu+1))\|\nabla f(x)\|^2 + \frac{\eta^2}{2}M\sigma^2$$

For the second line, using $G \preceq MI$ again, we have $\frac{\eta^2}{2}\lambda^2\delta^T G\delta \leq \eta^2\frac{M}{2}\lambda^2\|\delta\|^2$. Hence,

$$-\frac{\eta}{4}m\lambda\|\delta\|^2 + \frac{\eta^2}{2}\lambda^2\delta^T G\delta \leq -\frac{\eta}{4}\lambda(m - 2\eta M\lambda)\|\delta\|^2$$

For the third line, we apply the inequality $a^2 + b^2 \geq 2ab$ to obtain

$$\frac{\eta}{2}\|\nabla f(x)\|^2 + \frac{\eta}{4}m\lambda\|\delta\|^2 \geq \frac{\eta}{\sqrt{2}}\sqrt{m\lambda}\|\nabla f(x)\|\|\delta\|$$

On the other hand, the Cauchy-Schwarz inequality yields $\nabla f(x)^T G\delta = (G^{1/2}\nabla f(x))^T(G^{1/2}\delta) \leq$

$M\|\nabla f(x)\|\|\delta\|$. Hence

$$-\frac{\eta}{2}\|\nabla f(x)\|^2 - \frac{\eta}{4}m\lambda\|\delta\|^2 + \eta^2\lambda\nabla f(x)^T G\delta \leq -\frac{\eta\sqrt{\lambda}}{\sqrt{2}}(\sqrt{m} - \eta M\sqrt{2\lambda})\|\nabla f(x)\|\|\delta\|$$

Combining these inequalities yields the desired result. □

From this result, we can then demonstrate a sublinear convergence rate for LSGD, which holds whenever the leader is chosen so that its value is lower than the other workers.

**Theorem 4.5.2.** *Let $f$ satisfy Assumption 1. Suppose that the leader $z_k$ is always chosen so that $f(z_k) \leq f(x_k)$. If $\eta, \lambda$ are fixed so that $\eta \leq (2M(\nu+1))^{-1}$ and $\eta\lambda \leq (2\kappa)^{-1}, \eta\sqrt{\lambda} \leq (\kappa\sqrt{2m})^{-1}$, then*

$$\limsup_{k\to\infty} \mathbb{E}f(x_k) - f(x^*) \leq \frac{1}{2}\eta\kappa\sigma^2.$$

*If $\eta$ decreases at the rate $\eta_k = \Theta(\frac{1}{k})$, then $\mathbb{E}f(x_k) - f(x^*) = O(\frac{1}{k})$.*

*Proof.* This result follows (4.5.2) and Theorems 4.6 and 4.7 of [17]. □

The $O(\frac{1}{k})$ rate of LSGD matches that of comparable distributed methods. Both Hogwild [65] and EASGD achieve a rate of $O(\frac{1}{k})$ on strongly convex objective functions. We note that convergence rates have not been analyzed for many distributed algorithms (including DOWNPOUR [68] and Parle [66]).

One may ask whether LSGD may *surpass* the $\frac{1}{k}$ sublinear convergence rate on strongly concex functions. However, the $\Omega(\frac{1}{k})$ lower bound obtained in [84] also applies to LSGD, since we may view each LSGD iteration as making $p$ calls to a stochastic first-order oracle.

### 4.6 Stochastic Convex Optimization with Communication Delay

In practice, communication between distributed machines is costly. The LSGD algorithm has a *communication period* $\tau$ for which the leader is only updated every $\tau$ iterations, so each node can run independently during that period. This $\tau$ is allowed to differ between nodes, and over time,

which captures the asynchronous and multi-leader variants of LSGD. We write $x_{k,j}$ for the $j$-th step during the $k$-th period. It may occur that $f(z) > f(x_{k,j})$ for some $k, j$, that is, the current solution $x_{k,j}$ is now better than the last selected leader. In this case, the leader term $\lambda(x - z)$ may no longer be beneficial, and instead simply pulls $x$ toward $z$. There is no general way to determine how many steps are taken before this event. However, once $f(x) \leq f(z)$, we can show that subsequent LSGD steps will not make the solution *worse* than the stale leader $z$, up to gradient noise. This is captured by the following corollary:

**Corollary 4.6.1.** *If $f(x) \leq f(z)$, then*

$$\begin{aligned}\mathbb{E}f(x_+) \leq f(z) + \frac{\eta^2}{2}M\sigma^2 \\ - \frac{\eta}{2}(1 - \eta M(\nu + 1))\|\nabla f(x)\|^2 - \frac{\eta}{4}\lambda(m - 2\eta M\lambda)\|\delta\|^2 \\ - \frac{\eta\sqrt{\lambda}}{\sqrt{2}}(\sqrt{m} - \eta M\sqrt{2\lambda})\|\nabla f(x)\|\|\delta\|\end{aligned}$$

*Proof.* Follows from Lemma 4.5.1. □

**Corollary 4.6.2.** *In the deterministic case, once we reach a point with $f(x) \leq f(z)$, then $f(x_+) \leq f(z)$ as well.*

Suppose we simply continue to run the LSGD algorithm with fixed $z$. As $\tau$ goes to infinity, LSGD converges to the minimizer of $\psi(x) = f(x) + \frac{\lambda}{2}\|x - z\|^2$, which is quantifiably better than $z$ as captured in Lemma 4.6.3. Together, these facts show that LSGD is safe to use with long communication periods as long as the original leader is good.

**Lemma 4.6.3.** *Let $f$ be twice differentiable and strongly convex, with $mI \preceq \nabla^2 f(x)$ for $m > 0$, and let $x^*$ be the minimizer of $f$. Fix a constant $\lambda$ and any point $z$, and define the function $\psi(x) = f(x) + \frac{\lambda}{2}\|x - z\|^2$. Since $\psi$ is strongly convex, it has a unique minimizer $w$. The minimizer $w$ satisfies*[2]

$$f(w) - f(x^*) \leq \frac{\lambda}{m + \lambda}(f(z) - f(x^*)) \tag{4.6.1}$$

[2]If we also assume that $f$ is Lipschitz differentiable (that is, $\nabla^2 f(x) \preceq MI$), then we can obtain a similar inequality to the second directly from the first, but this is generally weaker than the bound given here.

*and*

$$\|w - x^*\|^2 \leq \frac{\lambda^2}{m(m+\lambda)}\|z - x^*\|^2 \tag{4.6.2}$$

*Proof.* The first-order condition for $w$ implies that $\nabla f(w) + \lambda(w - z) = 0$, so $\lambda^2\|w - z\|^2 = \|\nabla f(w)\|^2$. Combining this with the Polyak-Łojasiewicz inequality, we obtain

$$\frac{\lambda}{2}\|w - z\|^2 = \frac{1}{2\lambda}\|\nabla f(w)\|^2 \geq \frac{m}{\lambda}(f(w) - f(x^*))$$

We have $\psi(w) \leq \psi(z) = f(z)$, so $f(w) - f(x^*) \leq f(z) - f(x^*) - \frac{\lambda}{2}\|w - z\|^2$. Substituting, $f(w) - f(x^*) \leq f(z) - f(x^*) - \frac{m}{\lambda}(f(w) - f(x^*))$, which yields the first inequality.

We also have $\psi(w) = f(w) + \frac{\lambda}{2}\|w - z\|^2 \leq \psi(x^*) = f(x^*) + \frac{\lambda}{2}\|x^* - z\|^2$, whence $f(w) - f(x^*) \leq \frac{\lambda}{2}(\|x^* - z\|^2 - \|w - z\|^2)$. Hence, we have

$$\begin{aligned} f(w) - f(x^*) &\leq \frac{\lambda}{2}(\|x^* - z\|^2 - \|w - z\|^2) \\ &\leq \frac{\lambda}{2}\|z - x^*\|^2 - \frac{m}{\lambda}(f(w) - f(x^*)) \end{aligned}$$

so $f(w) - f(x^*) \leq \frac{\lambda^2}{2(m+\lambda)}\|z - x^*\|^2$. Finally, $f(w) - f(x^*) \geq \frac{m}{2}\|w - x^*\|^2$, which yields the result. □

The theoretical results here and in Section 4.5 address two fundamental instances of the LSGD algorithm: the 'synchronous' case where communication occurs each round, and the 'infinitely asynchronous' case where communication periods are arbitrarily long. For unknown periods $\tau > 1$, it is difficult to demonstrate general quantifiable improvements beyond Corollary 4.6.1.

## 4.7 Stochastic Leader Selection

We analyze the impact of selecting the leader with errors. In practice, it is often costly to evaluate $f(x)$, as in deep learning. Instead, we estimate the values $f(x^{(i)})$, and then select $z$ as the variable having the smallest estimate.

Formally, suppose that we have an unbiased estimator $\widetilde{f}(x)$ of $f(x)$, with uniformly bounded

variance. At each step, a single sample $y_1, \ldots, y_p$ is drawn from each estimator $\widetilde{f}(x^{(1)}), \ldots, \widetilde{f}(x^{(p)})$, and then $z = \{x^{(i)} : y_i = \min\{y_1, \ldots, y_p\}\}$. We refer to this as *stochastic leader selection*.

We first bound the probability of selecting an incorrect leader. The result rests on the following technical lemma. We state it in terms of general random variables; in the context of LSGD leaders, we have $\mu_i = f(x_k^{(i)})$ and $Y_i = \widetilde{f}(x_k^{(i)})$, and select the leader $z = \widetilde{\mu}$.

**Lemma 4.7.1.** *Let $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_p$. Suppose that $Y_1, \ldots, Y_p$ is a collection of independent random variables with $\mathbb{E}Y_i = \mu_i$ and $\mathrm{Var}(Y_i) \leq \sigma^2$. Let $\widetilde{\mu} = \mu_m$ where $m = \mathrm{argmin}\{Y_1, \ldots, Y_p\}$. Then*

$$\Pr(\widetilde{\mu} \geq \mu_k) \leq 4\sigma^2 \sum_{i=k}^{p} \frac{1}{(\mu_i - \mu_1)^2}$$

*Therefore, for any $a \geq 0$,*

$$\Pr(\widetilde{\mu} \geq \mu_1 + a) \leq 4\sigma^2 \frac{p}{a^2}.$$

*Proof.* In order for $\mu_m \geq \mu_k$, we must have $Y_j \leq Y_1$ for some $j \geq k$. Thus, $\{\widetilde{\mu} \geq \mu_k\}$ is a subset of the event $\{Y_1 \geq \min\{Y_k, \ldots, Y_p\}\}$. Taking the union bound,

$$\Pr(Y_1 \geq \min\{Y_k, \ldots, Y_p\}) \leq \sum_{i=k}^{p} \Pr(Y_1 \geq Y_i)$$

Applying Chebyshev's inequality to $Y_1 - Y_i$, and noting that $\mathrm{Var}(Y_1 - Y_i) \leq 4\sigma^2$, we have

$$\Pr(Y_1 - Y_i \geq 0) \leq \Pr(|Y_1 - Y_i - (\mu_i - \mu_1)| \geq \mu_i - \mu_1) \leq \frac{4\sigma^2}{(\mu_i - \mu_1)^2}$$

□

Using this, we can obtain a bound on $\mathbb{E}\widetilde{\mu}$.

**Lemma 4.7.2.** *Let $\widetilde{\mu}$ be defined as in Lemma 4.7.1. Then*

$$\mathbb{E}\widetilde{\mu} - \mu_1 \leq 4\sqrt{p}\sigma$$

*Proof.* Recall that the expected value of a nonnegative random variable $Z$ can be expressed as

$\mathbb{E}Z = \int_0^\infty \Pr(Z \geq t)dt$. We apply this to the variable $\widetilde{\mu} - \mu_1$. Using Lemma 4.7.1, we obtain, for any $a > 0$,

$$\begin{aligned}\mathbb{E}\widetilde{\mu} - \mu_1 = \int_0^\infty \Pr(\widetilde{\mu} - \mu_1 \geq t)dt &= \int_0^a \Pr(\widetilde{\mu} - \mu_1 \geq t)dt + \int_a^\infty \Pr(\mu^* - \mu_1 \geq t)dt \\ &\leq a + \int_a^\infty \Pr(\widetilde{\mu} - \mu_1 \geq t)dt \\ &\leq a + \int_a^\infty 4\sigma^2 \frac{p}{t^2} dt = a + 4\sigma^2 \frac{p}{a}\end{aligned}$$

The AM-GM inequality implies that $a + 4\sigma^2 \frac{p}{a} \geq 4\sqrt{p}\sigma$, with equality when $a = 2\sqrt{p}\sigma$. □

**Corollary 4.7.3.** *Assume without loss of generality that $f(x^{(1)}) \leq \ldots \leq f(x^{(p)})$. Suppose that we have unbiased estimators $\widetilde{f}(x^{(1)}), \ldots, \widetilde{f}(x^{(p)})$ for the true function values, with uniformly bounded variance $\mathrm{Var}(\widetilde{f}(x^{(i)}) \leq \sigma_f^2$. Then the stochastic leader satisfies*

$$\mathbb{E}f(z) \leq f(x^{(1)}) + 4\sqrt{p}\sigma_f.$$

As a corollary of Corollary 4.7.3, we can show that stochastic leader selection has the effect of increasing the limiting variance of LSGD.

**Proposition 4.7.4.** *Suppose that LSGD has a gradient estimator with $\mathrm{Var}(\widetilde{g}(x)) \leq \sigma^2 + \nu\|\nabla f(x)\|^2$ and function estimator with $\sup_x \mathrm{Var}(\widetilde{f}(x)) \leq \sigma_f^2$. Then, taking the expectation with respect to the gradient estimator and the approximate leader, we have*

$$\begin{aligned}\mathbb{E}f(x_+) \leq f(x) &+ 4\eta\lambda\sqrt{p}\sigma_f + \frac{\eta^2}{2}M\sigma^2 \\ &- \frac{\eta}{2}(1 - \eta M(\nu + 1))\|\nabla f(x)\|^2 - \frac{\eta}{4}\lambda(m - 2\eta M\lambda)\|\delta\|^2 \\ &- \frac{\eta\sqrt{\lambda}}{\sqrt{2}}(\sqrt{m} - \eta M\sqrt{2\lambda})\|\nabla f(x)\|\|\delta\|\end{aligned}$$

*Proof.* From Lemma 4.5.1, we obtain

$$\begin{aligned}\mathbb{E}f(x_+) \leq f(x) &- \frac{\eta}{2}(1-\eta M(\nu+1))\|\nabla f(x)\|^2 \\ &- \frac{\eta}{4}\lambda(m-2\eta M\lambda)\|\delta\|^2 \\ &- \frac{\eta\sqrt{\lambda}}{\sqrt{2}}(\sqrt{m}-\eta M\sqrt{2\lambda})\|\nabla f(x)\|\|\delta\| \\ &- \eta\lambda(f(x)-\mathbb{E}f(z)) + \frac{\eta^2}{2}M\sigma^2\end{aligned}$$

Note that in the last line, we have $\mathbb{E}f(z)$ because $z$ is now stochastic. By Corollary 4.7.3, $\mathbb{E}f(z) \leq \mu_1 + 4\sqrt{p}\sigma_f$, where $\mu_1 \leq f(x)$. Hence $f(x) - \mathbb{E}f(z) \geq f(x) - \mu_1 - 4\sqrt{p}\sigma_f \geq -4\sqrt{p}\sigma_f$, and so $-\eta\lambda(f(x)-\mathbb{E}f(z)) \leq 4\eta\lambda\sqrt{p}\sigma_f$. □

Observe that the effect of the stochastic leader is an increase of $4\eta\lambda\sqrt{p}\sigma_f$ in the constant error term. Since the new additive error is of order $\eta$ rather than $\eta^2$, we cannot guarantee convergence with $\eta_k = \Theta(\frac{1}{k})$, unless $\lambda_k$ is also decreasing[3]. By a similar analysis as Theorem 4.5.2, we obtain the following for LSGD with stochastic leader selection:

**Theorem 4.7.5.** *Let $f$ satisfy Assumption 1. Suppose that LSGD has a gradient estimator with $\mathrm{Var}(\widetilde{g}(x)) \leq \sigma^2 + \nu\|\nabla f(x)\|^2$ and function estimator with $\sup_x \mathrm{Var}(\widetilde{f}(x)) \leq \sigma_f^2$.*

*If $\eta, \lambda$ are fixed so that $\eta \leq (2M(\nu+1))^{-1}$ and $\eta\lambda \leq (2\kappa)^{-1}, \eta\sqrt{\lambda} \leq (\kappa\sqrt{2m})^{-1}$, then*

$$\limsup_{k\to\infty} \mathbb{E}f(x_k) - f(x^*) \leq \frac{1}{2}\eta\kappa\sigma^2 + \frac{4}{m}\lambda\sqrt{p}\sigma_f.$$

*If $\eta, \lambda$ decrease at the rate $\eta_k = \Theta(\frac{1}{k}), \lambda_k = \Theta(\frac{1}{k})$, then $\mathbb{E}f(x_k) - f(x^*) = O(\frac{1}{k})$.*

*Proof.* Interpret the term $4\eta\lambda\sqrt{p}\sigma_f$ as additive noise. Note that if $\eta_k, \lambda_k = \Theta(\frac{1}{k})$, then $\eta\lambda = \Theta(\frac{1}{k^2})$. The proof is then similar to Theorem 4.5.2 and follows from Theorems 4.6 and 4.7 of [17]. □

An unfortunate fact is that the error is of order $O(\sqrt{p})$, which grows as the number of workers increases. Interestingly, it turns out that error of order $\Omega(\sqrt{p})$ is tight for the problem of selecting

[3]For intuition, note that $\sum_{n=1}^{\infty} \frac{1}{n}$ is divergent.

a minimizer from a single estimation of random variables (Lemma 4.7.2).

**Proposition 4.7.6.** *For each $p \geq 2$ and all $\sigma > 0$, there exists $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_p$ and unbiased estimators $Y_i$ of $\mu_i$ with $\mathrm{Var}(Y_i) \leq \sigma^2$ such that*

$$\mathbb{E}\widetilde{\mu} - \mu_1 \geq (1 - \exp(-1/12))\sqrt{p}\sigma$$

*Proof.* Let $\mu_1 = 0$ and $\mu_2 = \ldots = \mu_p = \sqrt{p}\sigma$. Let $Y_1 = 0$ with probability 1 and $Y_2, \ldots, Y_p$ i.i.d with the following 3-point distribution:

$$\begin{cases} -\sigma & \text{with probability } \frac{1}{6p} \\ \sqrt{p}\sigma & \text{with probability } 1 - \frac{1}{3p} \\ 2\sqrt{p}\sigma + \sigma & \text{with probability } \frac{1}{6p} \end{cases}$$

It is easy to verify that $\mathbb{E}Y_i = \sqrt{p}\sigma$ and $\mathrm{Var}(Y_i) = \frac{\sigma^2}{3p}(\sqrt{p}+1)^2 \leq \sigma^2$ (when $p \geq 2$).

We have $\widetilde{\mu} = \sqrt{p}\sigma$ if any of $Y_2, \ldots, Y_p$ takes the value $-\sigma$. Considering the complement, we have

$$\Pr(Y_2, \ldots, Y_p \geq \sqrt{p}\sigma) = \left(1 - \frac{1}{6p}\right)^{p-1} \leq \exp\left(-\frac{1}{6p}(p-1)\right) \leq \exp(-1/12)$$

Thus, $\Pr(\widetilde{\mu} = \sqrt{p}\sigma) \geq 1 - \exp(-1/12)$ and we obtain $\mathbb{E}\widetilde{\mu} - \mu_1 \geq (1 - \exp(-1/12))\sqrt{p}\sigma$. □

We note that this lower bound is only effective when $\sqrt{p}\sigma$ is smaller than the largest difference $\mu_p - \mu_1$. In particular, the construction of the counterexample no longer holds if we require that $\mu_p \leq B$ uniformly for all $p$. This may occur for stochastic leader selection if it is known *a priori* that the objective function is bounded, e.g. $|f(x)| \leq B$ for all $x \in \mathbb{R}^n$. Clearly we then have upper bounds on the possible values of the estimator $Y_i = \widetilde{f}(x^{(i)})$ which take precedence when $\sqrt{p}\sigma_f \geq B$.

## 4.8 Nonconvex Optimization

In this section, we consider nonconvex optimization with the deterministic LGD. We first show that LGD avoids the 'spurious' local minimizer problem that affects EASGD. This is formalized in Proposition 4.8.2.

For each $i$, let $\Omega_i = \{x \in \mathbb{R}^n : f(x^{(i)}) < \min f(x^{(j)}) : j \neq i\}\}$. That is, $\Omega_i$ is the set of points on which $x^{(i)}$ is the unique minimizer. Define $\Omega = \bigcup_{i=1}^{p} \Omega_i$.

**Proposition 4.8.1.** *$\Omega_i$ is open.*

*Proof.* This follows immediately from the continuity of $f$. □

**Proposition 4.8.2.** *Let $x^* = (w^{(1)}, \ldots, w^{(p)}) \in \Omega_i$ be a stationary point of the LGD objective function. Then $\nabla f^{(i)}(w^{(i)}) = 0$.*

*Proof.* This follows from the fact that on $\Omega_i$, $\frac{\partial}{\partial x^{(i)}} \mathcal{L}\big|_{w^{(i)}} = \nabla f^{(i)}(w^{(i)})$. □

Next, we consider the *deterministic* version of the algorithm and its properties for nonconvex functions. It can be shown that for the deterministic algorithm LGD with *any choice of finite communication periods*, there will always be some variable $x^{(i)}$ such that $\liminf \|\nabla f(x_k^{(i)})\| = 0$.

**Lemma 4.8.3.** *Let $f$ be Lipschitz differentiable, with Lipschitz constant $M$. If the gradient descent stepsize $\eta < \frac{2}{M}$, then $\|\nabla f(x)\|^2 \leq \alpha(f(x) - f(x^+))$, where $\alpha = \frac{2}{\eta(2-\eta M)}$.*

*Proof.* By Taylor expansion,

$$\begin{aligned} f(x^+) &= f(x) - \eta \nabla f(x)^T \nabla f(x) + \frac{\eta^2}{2} \nabla f(x)^T \widetilde{G} \nabla f(x) \\ &\leq f(x) - \eta \|\nabla f(x)\|^2 + \frac{\eta^2}{2} M \|\nabla f(x)\|^2 \\ &= f(x) - \frac{\eta}{2}(2 - \eta M) \|\nabla f(x)\|^2 \end{aligned}$$

Rearranging yields the desired result. □

**Lemma 4.8.4.** *Let $\widetilde{x}_k$ denote the leader at the end of the $k$-th period. If the LGD stepsize is chosen so that $\eta_i < M_i$, then $f(\widetilde{x}_k) \leq f(\widetilde{x}_{k-1})$.*

*Proof.* Assume that $\widetilde{x}_{k-1} = x_{k-1}^{(1)}$. Since $x^{(1)}$ is the leader during the $k$-th period, the LGD steps for $x^{(1)}$ are gradient descent steps. By Lemma 4.8.3, $\eta_1$ has been chosen so that gradient descent on $f^{(1)}$ is monotonically decreasing, so we know that $f^{(1)}(x_k^{(1)}) \leq f^{(1)}(x_{k-1}^{(1)})$. Hence $f(\widetilde{x}_k) \leq f^{(1)}(x_k^{(1)}) \leq f^{(1)}(x_{k-1}^{(1)}) = f(\widetilde{x}_{k-1})$. □

**Proposition 4.8.5.** *Assume that $f$ is bounded below and $M$-Lipschitz differentiable. If the LGD step sizes are selected so that $\eta_i < \frac{2}{M_i}$, then for every $i$ such that $x^{(i)}$ is the leader infinitely often,* $\liminf_k \|\nabla f(x_k^{(i)})\| = 0$.

*Proof.* Without loss of generality, we assume it to be $x^{(1)}$. Let $\tau(1), \tau(2), \ldots$ denote the periods where $x^{(1)}$ is the leader, with $b(k)$ steps in the period $\tau(k)$. By Lemma 4.8.4, $f(x_{\tau(k+1)}^{(1)}) \leq f(x_{\tau(k)}^{(1)})$, since the objective value of the leaders is monotonically decreasing. Now, by Lemma 4.8.3, we have $\sum_{i=0}^{b(k)-1} \|\nabla f(x_{\tau(k),i}^{(1)})\|^2 \leq \alpha(f(x_{\tau(k),0}^{(1)}) - f(x_{\tau(k),b(k)}^{(1)})) = \alpha(f(x_{\tau(k)}^{(1)}) - f(x_{\tau(k+1)}^{(1)}))$. Since $f$ is bounded below, and the sequence $\{f(x_{\tau(k)}^{(1)})\}$ is monotonically decreasing, we must have $f(x_{\tau(k)}^{(1)}) - f(x_{\tau(k+1)}^{(1)}) \to 0$. Therefore, we must have $\|\nabla f(x_{\tau(k),i}^{(1)})\| \to 0$.

Note that there necessarily exists an index $i$ such that $x^{(i)}$ is the leader infinitely often. □

It follows that the deterministic LGD algorithm is convergent in the following sense:

$$\min_{1 \leq i \leq p} \liminf_{k \to \infty} \|\nabla f(x_k^{(i)})\| = 0.$$

Interestingly, while we proved this result under the assumption of exact gradients and exact leaders, it holds under arbitrary finite communication delays.

## 4.9 Quantifiable Improvements of LGD

In this section, we discuss how LGD can obtain better search directions than gradient descent. In general, it is difficult to determine when the LGD step will satisfy $f(x - \eta(\nabla f(x) + \lambda(x - z))) \leq$

$f(x - \eta\nabla f(x))$, since this depends on the precise combination of $f, x, z, \eta, \lambda$, and moreover, the maximum allowable value of $\eta$ is different for LGD and gradient descent. Instead, we measure the goodness of a search direction by the angle it forms with the *Newton direction*

$$d_N(x) = -(\nabla^2 f(x))^{-1}\nabla f(x).$$

The Newton method is locally quadratically convergent around local minimizers with non-singular Hessian, and converges in a single step for quadratic functions if $\eta = 1$. Hence, we consider it desirable to have search directions that are close to $d_N$.

Let $\theta(u, v)$ denote the angle between $u, v$, which is the angle $\theta \in [0, \pi]$ such that $\cos(\theta(u, v)) = \frac{u^T v}{\|u\|\|v\|}$. Let $d_z = -(\nabla f(x) + \lambda(x - z))$ be the LGD direction[4] with leader $z$, and $d_G(x) = -\nabla f(x)$. The *angle improvement set* is the set of leaders

$$I_\theta(x, \lambda) = \{z : f(z) \leq f(x), \theta(d_z, d_N(x)) \leq \theta(d_G(x), d_N(x))\}$$

for which the angle between $d_z$ and $d_N$ is smaller than the angle between the gradient and $d_N$. The set of all candidate leaders is $E = \{z : f(z) \leq f(x)\}$. We aim to show that a large subset of leaders in $E$ belong to $I_\theta(x, \lambda)$.

In this section, we consider the positive definite quadratic $f(x) = \frac{1}{2}x^T Ax$ with condition number $\kappa$ and $d_G(x) = -Ax, d_N(x) = -x$. We use the $n$-dimensional volume $\mathrm{Vol}(\cdot)$ to measure the relative size of sets: an ellipsoid $E$ given by $E = \{x : x^T Ax \leq 1\}$ has volume $\mathrm{Vol}(E) = \det(A)^{-1/2}\mathrm{Vol}(S_n)$, where $S_n$ is the unit ball. We show that $\mathrm{Vol}(I_\theta(x, \lambda)) \geq \frac{1}{2}\mathrm{Vol}(E)$ in two settings: where $\lambda$ is small, and where $A$ is ill-conditioned.

First, we analyze $I_\theta(x, \lambda)$ with small $\lambda$, for arbitrary convex $A$. Define the *cone* with center $d$ and angle $\theta_c$ is defined to be

$$\mathrm{cone}(d, \theta_c) = \{x : x^T d \geq 0, \theta(x, d) \leq \theta_c\}.$$

[4]Note that we use $\eta = 1$ in the LGD step $d_z$ since we may equivalently rescale $\lambda$ when measuring angles.

We record the following facts about cones which will be useful.

**Proposition 4.9.1.** *Let $C \subseteq \mathrm{cone}(d, \theta_c)$. If $y$ is a point such that $sy \in C$ for some $s \geq 0$, then $y \in \mathrm{cone}(d, \theta_c)$.*

*Proof.* This follows immediately from the fact that $\theta(y, d) = \theta(sy, d)$ for all $s \geq 0$. □

**Proposition 4.9.2.** *Let $C = \mathrm{cone}(d, \theta_c)$ with $\theta_c > 0$. The outward normal vector at the point $x \in \partial C$ is given by $N_x = x - \frac{\|x\|}{\cos(\theta_c)\|d\|} d$. Moreover, if $v$ satisfies $N_x^T v < 0$, then for sufficiently small positive $\lambda$, $x + \lambda v \in \mathrm{cone}(d, \theta_c)$.*

*Proof.* The first statement follows from the second, by the supporting hyperplane theorem.

Write $\gamma = \cos(\theta_c)$. Let $N_x = x - \frac{\|x\|}{\gamma\|d\|} d$, and let $v$ be a unit vector with $N_x^T v = x^T v - \frac{\|x\|}{\gamma\|d\|} d^T v < 0$. The angle satisfies

$$\cos(\theta(x + \lambda v, d)) = \frac{d^T(x + \lambda v)}{\|d\|\|x + \lambda v\|} = \frac{d^T x + \lambda d^T v}{\|d\|\sqrt{\|x\|^2 + \lambda^2\|v\|^2 + 2\lambda x^T v}}$$

Differentiating, the numerator $g(\lambda)$ of $\frac{\partial}{\partial \lambda} \cos(\theta(x + \lambda v, d))$ is given by

$$g(\lambda) = \|x\|^2 v^T d - x^T v x^T d + \lambda \cdot (2v^T d x^T d + \|v\|^2(\lambda v - x)^T d - \lambda\|v\|^2 v^T d - x^T v v^T d)$$

Evaluating at $\lambda = 0$ and using $x^T v - \frac{\|x\|}{\gamma\|d\|} d^T v < 0$, we obtain

$$\begin{aligned} g(0) = \|x\|^2 v^T d - x^T v x^T d &= \|x\|^2 v^T d - x^T v(\gamma\|x\|\|d\|) \\ &= \|x\|(\|x\| v^T d - \gamma\|d\| x^T v) > 0. \end{aligned}$$

Therefore, for small positive $\lambda$, we have $\cos(\theta(x + \lambda v, d)) > \cos(\theta(x, d)) = \gamma$, so $x + \theta v \in \mathrm{cone}(d, \theta_c)$. □

We show that as $\lambda \to 0$, at least half of the level set $\{f(z) \leq f(x)\}$ belongs to $I_\theta(x, \lambda)$.

**Proposition 4.9.3.** *Let $x$ be any point such that $\theta_x = \theta(d_G(x), d_N(x)) > 0$, and let $E = \{z : f(z) \leq f(x)\}$. Let $C = \mathrm{cone}(-x, \theta_x)$, and let $N_x$ be the outward normal $-\nabla f(x) + \frac{\|\nabla f(x)\|}{\cos(\theta_x)\|x\|}x$ of the cone $C$ at the point $-\nabla f(x)$. Then*

$$\bigcup_{\lambda>0} I_\theta(x, \lambda) \supseteq E \cap \{z : N_x^T z < N_x^T x\} \tag{4.9.1}$$

*and consequently,* $\lim_{\lambda\to 0} \mathrm{Vol}(I_\theta(x, \lambda)) \geq \frac{1}{2}\mathrm{Vol}(E)$.

*Proof.* First, note that if $\lambda_2 \leq \lambda_1$, then for all $z$ with $-\nabla f(x) + \lambda_1 z \in C$, we also have $-\nabla f(x) + \lambda_2 z \in C$ by the convexity of $C$. Therefore $I_\theta(x, \lambda_2) \supseteq I_\theta(x, \lambda_1)$, so $\lim_{\lambda\to 0} \mathrm{Vol}(I_\theta(x, \lambda))$ exists. We first prove the second statement. For any normal vector $h$ and $\beta > 0$, $\mathrm{Vol}(E \cap \{z : h^T z < \beta\}) \geq \frac{1}{2}\mathrm{Vol}(E)$, since the center $0 \in \{z : h^T z < \beta\}$. The result follows because $N_x^T x > 0$.

To prove (4.9.1), observe that $z \in I_\theta(x, \lambda)$ if equivalent to $-\nabla f(x) + \lambda(z - x) \in \mathrm{cone}(-x, \theta_c)$. By Proposition 4.9.2, there exists $\lambda > 0$ with $-\nabla f(x) + \lambda(z - x) \in \mathrm{cone}(-x, \theta_c)$ if $N_x^T(z - x) < 0$. Hence, it follows that every point in $E \cap \{z : N^T z < N^T x\}$ is contained in $I_\theta(x, \lambda)$ for some $\lambda > 0$. □

Proposition 4.9.3 implies that many leaders $z$ improve the angle of the step direction when $\lambda$ is small. The case where we do not allow $\lambda$ to shrink is more difficult to analyze. However, for points where the gradient direction is close to orthogonal to the Newton direction, we can show that the entire half-space $E \cap \{z : x^T z \leq 0\} \subseteq I_\theta(x, \lambda)$ for any $\lambda$. These points where the gradient and Newton direction are 'near-orthogonal' are also precisely those points where using the leader direction $d_z$ may be most useful. When $A$ is well-conditioned and the gradient is already very close to the Newton direction[5], there is little benefit to using $d_z$. Hence, we consider ill-conditioned $A$.

For $r \geq 2$, define

$$S_r = \left\{x : \cos(\theta(d_G(x), d_N(x))) = \frac{r}{\sqrt{\kappa}}\right\}$$

[5]In particular, if $A = \alpha I$, we have $d_G(x) = d_N(x)$ at every point and no improvement is possible.

or equivalently,

$$S_r = \left\{ x : \tan(\theta(d_G(x), d_N(x))) = \sqrt{\frac{\kappa}{r^2} - 1} \right\}.^6$$

The set $S_r$ comprises the directions $x$ where the angle between the gradient and Newton steps is large. The next proposition shows that $S_r$ is nontrivial.

**Proposition 4.9.4.** *There exists a direction $x$ such that $\cos(\theta(d_G(x), d_N(x))) = 2(\sqrt{\kappa} + \sqrt{\kappa^{-1}})^{-1}$. Thus, for all $r \geq 2$, there exists a direction $x$ with $\cos(\theta(d_G(x), d_N(x))) \leq \frac{r}{\sqrt{\kappa}}$.*

*Proof.* Take $x = \sqrt{\frac{\alpha_n}{\alpha_1+\alpha_n}} e_1 + \sqrt{\frac{\alpha_1}{\alpha_1+\alpha_n}} e_n$. It is easy to verify that $\cos(\theta(d_G, d_N)) = 2(\sqrt{\kappa} + \sqrt{\kappa^{-1}})^{-1}$. □

**Proposition 4.9.5.** *For any $x$, let $\theta_x = \theta(d_G(x), d_N(x))$. We have*

$$\max\{\|z\|^2 : f(z) \leq f(x), z^T x = 0\} \leq \kappa \cos(\theta_x)\|x\|^2$$

*Proof.* Form the maximization problem

$$\begin{cases} \max\limits_{z} & z^T z \\ & z^T A z \leq x^T A x \\ & z^T x = 0 \end{cases}$$

The KKT conditions for this problem imply that the solution satisfies $z - \mu_1 A z - \mu_2 x = 0$, for Lagrange multipliers $\mu_1 \geq 0, \mu_2$. Since $z^T x = 0$, we obtain $z^T z = \mu_1 z^T A z$, and thus $\frac{1}{M} \leq \mu_1 \leq \frac{1}{m}$. Since $f(z) \leq f(x)$, we find that $z^T z \leq \frac{1}{m} x^T A x$. Using $\cos(\theta_x) = \frac{x^T A x}{\|x\|\|Ax\|}$, we obtain

$$z^T z \leq \frac{1}{m} \cos(\theta_x)\|x\|\|Ax\| \leq \kappa \cos(\theta_x)\|x\|^2.$$

□

We are now ready to prove that $I_\theta(x, \lambda)$ is large for particular $S_r$.

[6] Note that increasing $r$ corresponds to decreasing angles.

**Proposition 4.9.6.** *Let* $R_\kappa = \{r : \frac{r}{\sqrt{\kappa}} + \frac{r^{3/2}}{\kappa^{1/4}} \leq 1\}$. *Consider* $f(x) = \frac{1}{2}x^T Ax$ *with condition number* $\kappa$. *Let* $x \in S_r$ *for* $r \in R_\kappa$, *and let* $E = \{y : f(y) \leq f(x)\}$, $E_2 = \{z \in E : z^T x \leq 0\}$, $\theta_x = \theta(d_G(x), d_N(x))$. *Then for all* $z \in E_2$ *and any* $\lambda \geq 0$, *the LGD direction* $d_z = -(\nabla f(x) + \lambda(x - z))$ *satisfies* $\theta(d_z, d_N(x)) \leq \theta_x$. *Thus,* $E_2 \subseteq I_\theta(x, \lambda)$, *and therefore* $\mathrm{Vol}(I_\theta(x, \lambda)) \geq \mathrm{Vol}(E_2) = \frac{1}{2}\mathrm{Vol}(E)$.

*Proof.* Define $D_2 = \{z - x : z \in E_2\}$[7]. The set of possible LGD directions with $z \in E_2$ is given by $D_3 = \{-\nabla f(x) + \lambda\delta : \delta \in D_2, \lambda \geq 0\}$. Since $d_N(x) = -x$, our desired result is equivalent to $D_3 \subseteq \mathrm{cone}(-x, \theta_x)$.

Define the subset $D_2' = \{z - x : z \in E_2, x^T z = 0\}$. We claim that it suffices to prove that $D_2' \subseteq \mathrm{cone}(-x, \theta_x)$. To see this, consider any $\lambda\delta$ for $\lambda \geq 0$ and $\delta \in D_2$. We have $x^T(\lambda\delta) = \lambda x^T(z - x) \leq -\lambda x^T x < 0$, so there exists a scalar $s$ with $x^T(s\lambda\delta) = -x^T x$, whence $s\lambda\delta \in D_2' \subseteq \mathrm{cone}(-x, \theta_x)$. By Proposition 4.9.1, $\lambda\delta \in \mathrm{cone}(-x, \theta_x)$. Since $-\nabla f(x) \in \mathrm{cone}(-x, \theta_x)$, convexity implies that $-\nabla f(x)\lambda\delta \in \mathrm{cone}(-x, \theta_x)$. Thus, $D_2' \subseteq \mathrm{cone}(-x, \theta_x)$ implies that $D_3 \subseteq \mathrm{cone}(-x, \theta_x)$.

To complete the proof, let $\delta = z - x \in D_2'$ and observe that $\cos(\theta(\delta, d_N(x))) = \frac{x^T(x-z)}{\|x\|\|x-z\|}$. By Proposition 4.9.5 and the definition of $S_r$,

$$\max\{\|z\| : z \in E_2, z^T x = 0\} \leq \sqrt{\kappa}\sqrt{\cos(\theta_x)}\|x\| = \sqrt{r}\kappa^{1/4}\|x\|$$

We compute that

$$\begin{aligned} x^T(x - z) - \frac{r}{\sqrt{\kappa}}\|x\|\|x - z\| &\geq \|x\|^2 - \frac{r}{\sqrt{\kappa}}(\|x\|^2 + \|x\|\|z\|) \\ &\geq \|x\|^2 - \frac{r}{\sqrt{\kappa}}\|x\|^2 - \frac{r}{\sqrt{\kappa}}\|x\|(\sqrt{r}\kappa^{1/4}\|x\|) \\ &\geq \left(1 - \frac{r}{\sqrt{\kappa}} - \frac{r^{3/2}}{\kappa^{1/4}}\right)\|x\|^2 \geq 0 \end{aligned}$$

By the definition of $R_\kappa$, this is nonnegative, and thus $\theta(\delta, d_N(x)) \leq \theta_x$. This completes the proof. □

[7]Note the sign change from $x - z$ to $z - x$ here.

Note that Propositions 4.9.3 and 4.9.6 apply only to *convex* functions, or in the neighborhoods of local minimizers where the objective function is locally convex. In nonconvex landscapes, the Newton direction may point towards saddle points [85], which is undesirable; however, since Propositions 4.9.3 and 4.9.6 do not apply in this situation, these results do not imply that LSGD has harmful behavior. For nonconvex problems, our intuition is that many candidate leaders lie in directions of *negative curvature*, which would actually lead away from saddle points, but this is significantly harder to analyze since the set of candidates is unbounded a priori.

## 4.10 A Drawback of LSGD: Implicit Variance Reduction

Elastic Averaging SGD implicitly yields variance reduction when the number of workers $p$ increases. One example of this is [77, Corollary 3.1.1], which shows that when applied to a one-dimensional quadratic function, the EASGD consensus variable $\widetilde{x}$ has limiting mean-squared error of order $O(\frac{1}{p})$.

This is perhaps to be expected, since $\widetilde{x}$ is updated towards the average of the worker variables. We do not expect LSGD to have the same property, and indeed, we can construct a counterexample which shows that it does not. In Proposition 4.10.1, we show that when the algorithm hyperparameters $\eta, \lambda$ are fixed, there exists $\epsilon > 0$ such that a LSGD update $x_+ = x - \eta\nabla f(x) - \eta\lambda(x - z)$ never brings $x_+$ into the interval $(-\epsilon, \epsilon)$. It follows that regardless of the number of workers $p$, none of the worker parameters will enter a fixed interval around the minimizer, and hence the limiting variance is independent of the number of workers.

**Proposition 4.10.1.** *Let $f(x) = \frac{1}{2}x^2$. For any suitable choice of hyperparameters $\eta, \lambda$[8], there exists a gradient estimator $\widetilde{g}(x)$ and $\overline{\epsilon} > 0$ such that $|x_k^{(i)}| \geq \overline{\epsilon}$ for every worker $i$ and all iterations $k$. In particular, $\overline{\epsilon}$ is independent of the number of workers.*

*Proof.* Our strategy will be to exhibit an estimator $\widetilde{g}(x)$ and a threshold $\epsilon > 0$ with the property that for any point $x$ and leader $z$ with $f(z) \leq f(x)$, the updated point $x_+$ satisfies $|x_+| \geq \epsilon$. It follows

[8] In particular, when $\eta, \lambda$ are small.

that if we define $f_0 = \min\{|x_0^{(1)}|, \ldots, |x_0^{(p)}|\}$[9], then taking $\bar{\epsilon} = \min\{f_0, \epsilon\}$, we have $|x_k^{(i)}| \geq \bar{\epsilon}$ for all workers $i$ and all iterations $k$.

We define the estimator $\widetilde{g}(x)$ by

$$\widetilde{g}(x) = \begin{cases} \{\nabla f(x) - \sigma, \nabla f(x) + \sigma\} & \text{with equal probability, if } |x| \leq \alpha \\ \nabla f(x) & \text{if } |x| > \alpha \end{cases}$$

To define $\epsilon$ and $\alpha$, we assume that $\eta + 2\eta\lambda < 1$. Choose $\epsilon$ so that

$$0 < \epsilon \leq \frac{\sigma}{2} \frac{\eta(1 - \eta - 2\eta\lambda)}{1 - \eta - \eta\lambda}$$

and then take $\alpha = \frac{\epsilon}{1-\eta-2\eta\lambda}$. These values are chosen so that the inequalities $\epsilon + (1-\eta)\alpha \leq \eta\sigma$ and $\epsilon + \eta\lambda\alpha \leq \eta\sigma$ hold.

Suppose first that $x > \alpha$. Since $z \geq -x$, the update satisfies

$$\begin{aligned} x_+ = x - \eta(x + \lambda(x - z)) &\geq (1 - \eta - \eta\lambda)x + \eta\lambda(-x) \\ &> (1 - \eta - 2\eta\lambda)\alpha = \epsilon \end{aligned}$$

Similarly, if $x < -\alpha$, we obtain $x_+ < -\epsilon$.

Next, suppose that $0 \leq x \leq \alpha$. By definition of $\widetilde{g}$, either $\widetilde{g}(x) = x + \sigma$ or $\widetilde{g}(x) = x - \sigma$. When $\widetilde{g}(x) = x + \sigma$, using that $z \leq x$, we have

$$\begin{aligned} x_+ = x - \eta(x + \sigma + \lambda(x - z)) &\leq (1 - \eta - \eta\lambda)x - \eta\sigma + \eta\lambda x \\ &\leq (1 - \eta)\alpha - \eta\sigma \leq -\epsilon \end{aligned}$$

[9]Assume that $x_0^{(i)} \neq 0$ for the initial points, i.e. we do not pick the minimizer as an initial point.

When $\widetilde{g}(x) = x - \sigma$, using $z \geq -x \geq -\alpha$, we have

$$\begin{aligned} x_+ = x - \eta(x - \sigma + \lambda(x - z)) &= (1 - \eta - \eta\lambda)x + \eta\sigma + \eta\lambda z \\ &\geq \eta\sigma - \eta\lambda\alpha \geq \epsilon \end{aligned}$$

Similarly, if $-\alpha \leq x \leq 0$, we obtain $x_+ \leq -\epsilon$ when $\widetilde{g}(x) = x + \sigma$ and $x_+ \geq \epsilon$ when $\widetilde{g}(x) = x - \sigma$. This completes the proof. □

Note that [77, Corollary 3.1.1] requires that $\lambda \to 0$ as $p \to \infty$ (expressed in our notation), in order to maintain the convergence of $\widetilde{x}$. To see that the counterexample in Proposition 4.10.1 still holds when $\lambda \to 0$, observe that the constructed estimator $\widetilde{g}(x)$ for $\lambda_0$ yields the same bounds if the LSGD update is made with any $\lambda \leq \lambda_0$.

# Chapter 5: Solving Structured Problems with Multiaffine ADMM

## 5.1 Introduction

The *alternating direction method of multipliers* (ADMM) is an iterative method which, in its original form, solves linearly-constrained separable optimization problems with the following structure:

$$(P0) \quad \begin{cases} \inf\limits_{x,y} & f(x) + g(y) \\ & Ax + By - b = 0. \end{cases}$$

The *augmented Lagrangian* $\mathcal{L}$ of the problem $(P0)$, for some *penalty parameter* $\rho > 0$, is defined to be

$$\mathcal{L}(x, y, w) = f(x) + g(y) + \langle w, Ax + By - b\rangle + \frac{\rho}{2}\|Ax + By - b\|^2.$$

In iteration $k$, with the iterate $(x^{(k)}, y^{(k)}, w^{(k)})$, ADMM takes the following steps:

1. Minimize $\mathcal{L}(x, y^{(k)}, w^{(k)})$ with respect to $x$ to obtain $x^{(k+1)}$.
2. Minimize $\mathcal{L}(x^{(k+1)}, y, w^{(k)})$ with respect to $y$ to obtain $y^{(k+1)}$.
3. Set $w^{(k+1)} \leftarrow w^{(k)} + \rho(Ax^{(k+1)} + By^{(k+1)} - b)$.

ADMM was first proposed [86, 87] for solving variational problems, and was subsequently applied to convex optimization problems with two blocks as in $(P0)$. Several techniques can be used to analyze this case, including an operator-splitting approach [88, 89, 90]. The survey articles [91, 39] provide convergence proofs from several viewpoints, and discuss numerous applications of ADMM. More recently, there has been considerable interest in extending ADMM convergence guarantees when solving problems with *multiple blocks* and *nonconvex* objective functions.

ADMM directly extends to the problem

$$(P1) \quad \begin{cases} \inf\limits_{x_1,x_2,\ldots,x_n} & f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) \\ & A_1x_1 + A_2x_2 + \ldots + A_nx_n - b = 0 \end{cases}$$

by minimizing $\mathcal{L}(x_1, \ldots, x_n, w)$ with respect to $x_1, x_2, \ldots, x_n$ successively. The multiblock problem turns out to be significantly different from the classical 2-block problem, even when the objective function is convex; for example, [92] exhibits an example with $n = 3$ blocks and $f_1, f_2, f_3 \equiv 0$ for which ADMM diverges for any value of $\rho$. Under certain conditions, the unmodified 3-block ADMM does converge. In [93], it is shown that if $f_3$ is strongly convex with condition number $\kappa \in [1, 1.0798)$ (among other assumptions), then 3-block ADMM is globally convergent. If $f_1, \ldots, f_n$ are all strongly convex, and $\rho > 0$ is sufficiently *small*, then [94] shows that multiblock ADMM is convergent. Other works along these lines include [95, 96, 97].

In the absence of strong convexity, modified versions of ADMM have been proposed that can accommodate multiple blocks. In [98] a new type of 3-operator splitting is introduced that yields a convergent 3-block ADMM (see also [99] for a proof that a 'lifting-free' 3-operator extension of Douglas-Rachford splitting does not exist). Convergence guarantees for multiblock ADMM can also be achieved through variants such as proximal ADMM, majorized ADMM, linearized ADMM [100, 101, 102, 103, 104, 105], and proximal Jacobi ADMM [102, 106, 107].

ADMM has also been extended to problems with *nonconvex* objective functions. In [108], it is proved that ADMM converges when the problem $(P1)$ is either a nonconvex *consensus* or *sharing* problem, and [109] proves convergence under more general conditions on $f_1, \ldots, f_n$ and $A_1, \ldots, A_n$. Proximal ADMM schemes for nonconvex, nonsmooth problems are considered in [110, 111, 112, 105]. More references on nonconvex ADMM, and comparisons of the assumptions used, can be found in [109].

In all of the work mentioned above, the system of constraints $C(x_1, \ldots, x_n) = 0$ is assumed to be linear. Consequently, when all variables other than $x_i$ have fixed values, $C(x_1, \ldots, x_n)$ becomes an *affine* function of $x_i$. However, this holds for more general constraints $C(\cdot)$ in the much larger

class of *multiaffine* maps (see Section 5.2). Thus, it seems reasonable to expect that ADMM would behave similarly when the constraints $C(x_1, \ldots, x_n) = 0$ are permitted to be multiaffine. To be precise, consider a more general problem than $(P1)$ of the form

$$(P2) \quad \begin{cases} \inf\limits_{x_1, x_2, \ldots, x_n} & f(x_1, \ldots, x_n) \\ & C(x_1, \ldots, x_n) = 0. \end{cases}$$

The augmented Lagrangian for $(P2)$ is

$$\mathcal{L}(x_1, \ldots, x_n, w) = f(x_1, \ldots, x_n) + \langle w, C(x_1, \ldots, x_n) \rangle + \frac{\rho}{2} \|C(x_1, \ldots, x_n)\|^2,$$

and ADMM for solving this problem is specified in Algorithm 4.

**Algorithm 4** ADMM

**Input:** $(x_1^0, \ldots, x_n^0), w^0, \rho$
**for** $k = 0, 1, 2, \ldots$ **do**
  **for** $i = 1, \ldots, n$ **do**
Compute $x_i^{(k+1)} \in \operatorname{argmin}_{x_i} \mathcal{L}(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \ldots, x_n^{(k)}, w^{(k)})$
  **end for**
$w^{(k+1)} \leftarrow w^{(k)} + \rho C(x_1^{(k+1)}, \ldots, x_n^{(k+1)})$
**end for**

While many problems can be modeled with multiaffine constraints, existing work on ADMM for solving multiaffine constrained problems appears to be limited. Boyd et al. [39] propose solving the nonnegative matrix factorization problem formulated as a problem with biaffine constraints, i.e.,

$$(\text{NMF1}) \quad \begin{cases} \inf\limits_{Z, X, Y} & \frac{1}{2}\|Z - B\|^2 \\ & Z = XY, X \geq 0, Y \geq 0, \end{cases}$$

by applying ADMM with alternating minimization on the blocks $Y$ and $(X, Z)$. The convergence of ADMM employed to solve the (NMF1) problem appears to have been an open question until a proof was given in [113][1]. A method derived from ADMM has also been proposed for optimizing

[1] [113] shows that every limit point of ADMM for the problem (NMF) is a constrained stationary point, but does not show that such limit points necessarily exist.

a biaffine model for training deep neural networks [114]. For general nonlinear constraints, a framework for "monitored" Lagrangian-based multiplier methods was studied in [115].

In this paper, we establish the convergence of ADMM for a broad class of problems with multiaffine constraints. Our assumptions are similar to those used in [109] for nonconvex ADMM; in particular, we do not make any assumption about the iterates generated by the algorithm. Hence, these results extend the applicability of ADMM to a larger class of problems which naturally have multiaffine constraints. Moreover, we prove several results about ADMM in Section 5.6 that hold in even more generality, and thus may be useful for analyzing ADMM beyond the setting considered here.

### 5.1.1 Organization of this paper

In Section 5.2, we define multilinear and multiaffine maps, and specify the precise structure of the problems that we consider. In Section 5.3, we provide several examples of problems that can be formulated with multiaffine constraints. In Section 5.4, we state our assumptions and main results (i.e., Theorems 5.4.1, 5.4.3 and 5.4.5). In Section 5.5, we present a collection of necessary technical material. In Section 5.6, we prove several results about ADMM that hold under weak conditions on the objective function and constraints. Finally, in Section 5.7, we complete the proof of our main convergence theorems (Theorems 5.4.1, 5.4.3 and 5.4.5), by applying the general techniques developed in Section 5.6. **??** contains proofs of technical lemmas. Section 5.8 presents an alternative biaffine formulation for deep neural network training. Section 5.9 presents additional formulations of problems where all ADMM subproblems have closed-form solutions.

### 5.1.2 Notation and Definitions

We consider only finite-dimensional real vector spaces. The symbols $\mathbb{E}, \mathbb{E}_1, \ldots, \mathbb{E}_n$ denote finite-dimensional Hilbert spaces, equipped with inner products $\langle \cdot, \cdot \rangle$. By default, we use the standard inner product on $\mathbb{R}^n$ and the trace inner product $\langle X, Y \rangle = \mathrm{Tr}(Y^T X)$ on the matrix space. Unless otherwise specified, the norm $\| \cdot \|$ is always the induced norm of the inner product. When

$A$ is a matrix or linear map, $\|A\|_{op}$ denotes the $L_2$ operator norm, and $\|A\|_*$ denotes the nuclear norm (the sum of the singular values of $A$). Fixed bases are assumed, so we freely use various properties of a linear map $A$ that depend on its representation (such as $\|A\|_{op}$), and view $A$ as a matrix as required.

For $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, the *effective domain* $\mathrm{dom}(f)$ is the set $\{x : f(x) < \infty\}$. The image of a function $f$ is denoted by $\mathrm{Im}(f)$. Similarly, when $A$ is a linear map represented by a matrix, $\mathrm{Im}(A)$ is the column space of $A$. We use $\mathrm{Null}(A)$ to denote the null space of $A$. The orthogonal complement of a linear subspace $U$ is denoted $U^\perp$.

To distinguish the derivatives of *smooth* (i.e., continuously differentiable) functions from subgradients, we use the notation $\nabla_X$ for partial differentiation with respect to $X$, and reserve the symbol $\partial$ for the set of *general subgradients* (Section 5.5.1); hence, the use of $\nabla f$ serves as a reminder that $f$ is assumed to be smooth. A function $f$ is *Lipschitz differentiable* if it is differentiable and its gradient is Lipschitz continuous.

When $\mathcal{X}$ is a tuple of variables $\mathcal{X} = (X_0, \ldots, X_n)$, we write $\mathcal{X}_{\neq \ell}$ for $(X_i : i \neq \ell)$. Similarly, $\mathcal{X}_{>\ell}$ and $\mathcal{X}_{<\ell}$ represent $(X_i : i > \ell)$ and $(X_i : i < \ell)$ respectively.

We use the term *constrained stationary point* for a point satisfying necessary first-order optimality conditions; this is a generalization of the Karush-Kuhn-Tucker (KKT) necessary conditions to nonsmooth problems. For the problem $\min_x\{f(x) : C(x) = 0\}$, where $C$ is smooth and $f$ possesses general subgradients, $x^*$ is a constrained stationary point if $C(x^*) = 0$ and there exists $w^*$ with $0 \in \partial f(x^*) + \nabla C(x^*)^T w^*$.

## 5.2 Multiaffine Constrained Problems

The central objects of this paper are multilinear and multiaffine maps, which generalize linear and affine maps.

**Definition 5.2.1.** *A map $\mathcal{M} : \mathbb{E}_1 \oplus \ldots \oplus \mathbb{E}_n \to \mathbb{E}$ is* multilinear *if, for all $i \leq n$ and all points*

$(\overline{X}_1, \ldots, \overline{X}_{i-1}, \overline{X}_{i+1}, \ldots, \overline{X}_n) \in \bigoplus_{j \neq i} E_j$, *the map* $\mathcal{M}_i : \mathbb{E}_i \to \mathbb{E}$ *given by*

$$X_i \mapsto \mathcal{M}(\overline{X}_1, \ldots, \overline{X}_{i-1}, X_i, \overline{X}_{i+1}, \ldots, \overline{X}_n)$$

*is* linear. *Similarly,* $\mathcal{M}$ *is* multiaffine *if the map* $\mathcal{M}_i$ *is* affine *for all* $i$ *and all points of* $\bigoplus_{j \neq i} \mathbb{E}_j$. *In particular, when* $n = 2$, *we say that* $\mathcal{M}$ *is* bilinear/biaffine.

We consider the convergence of ADMM for problems of the form:

$$(P) \quad \begin{cases} \inf\limits_{\mathcal{X}, \mathcal{Z}} & \phi(\mathcal{X}, \mathcal{Z}) \\ & A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) = 0, \end{cases}$$

where $\mathcal{X} = (X_0, \ldots, X_n)$, $\mathcal{Z} = (Z_0, \mathcal{Z}_>)$, $\mathcal{Z}_> = (Z_1, Z_2)$,

$$\phi(\mathcal{X}, \mathcal{Z}) = f(\mathcal{X}) + \psi(\mathcal{Z})$$

$$\text{and } \; A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) = \begin{bmatrix} A_1(\mathcal{X}, Z_0) + Q_1(Z_1) \\ A_2(\mathcal{X}) + Q_2(Z_2) \end{bmatrix}$$

with $A_1$ and $A_2$ being multiaffine maps and $Q_1$ and $Q_2$ being linear maps. The augmented Lagrangian $\mathcal{L}(\mathcal{X}, \mathcal{Z}, \mathcal{W})$, with penalty parameter $\rho > 0$, is given by

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}, \mathcal{W}) = \phi(\mathcal{X}, \mathcal{Z}) + \langle \mathcal{W}, A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) \rangle + \frac{\rho}{2} \| A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) \|^2,$$

where $\mathcal{W} = (W_1, W_2)$ are Lagrange multipliers.

We prove that Algorithm 4 converges to a constrained stationary point under certain assumptions on $\phi, A$, and $Q$, which are described in Section 5.4. Moreover, since the constraints are nonlinear, there is a question of constraint qualifications, which we address in Lemma 5.5.4.

We adopt the following notation in the context of ADMM. The variables in the $k$-th iteration are denoted $\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)}$ (with $X_i^{(k)}, Z_i^{(k)}, W_i^{(k)}$ for the $i$-th variable in each component). When analyzing a single iteration, the index $k$ is omitted, and we write $X = X^{(k)}$ and $X^+ = X^{(k+1)}$.

Similarly, we write $\mathcal{L}^{(k)} = \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ and will refer to $\mathcal{L} = \mathcal{L}^{(k)}$ and $\mathcal{L}^+ = \mathcal{L}^{(k+1)}$ for values within a single iteration.

## 5.3 Examples of Applications

In this section, we describe several problems with multiaffine constraints, and show how they can be formulated and solved by ADMM. Many important applications of ADMM involve introducing auxiliary variables so that all subproblems have closed-form solutions; we describe several such reformulations in section 5.9 that have this property.

### 5.3.1 Representation Learning

Given a matrix $B$ of data, it is often desirable to represent $B$ in the form $B = X * Y$, where $*$ is a bilinear map and the matrices $X, Y$ have some desirable properties. Two important applications follow:

1. *Nonnegative matrix factorization* (NMF) [116, 117] expresses $B$ as a product of nonnegative matrices $X \geq 0, Y \geq 0$.

2. *Inexact dictionary learning* (DL) [118] expresses every element of $B$ as a sparse combination of *atoms* from a *dictionary* $X$. It is typically formulated as

$$\text{(DL)} \quad \left\{ \inf_{X,Y} \quad \iota_S(X) + \|Y\|_1 + \tfrac{\mu}{2}\|XY - B\|^2, \right.$$

   where $\iota_S$ is the indicator function for the set $S$ of matrices whose columns have unit $L_2$ norm, and here $\|Y\|_1$ is the entrywise 1-norm $\sum_{i,j} |Y_{ij}|$. The parameter $\mu$ is an input that sets the balance between trying to recover $B$ with high fidelity versus finding $Y$ with high sparsity.

Problems of this type can be modeled with bilinear constraints. As already mentioned in Section 5.1, [39, 113] propose the bilinear formulation (NMF1) for nonnegative matrix factorization.

The inexact dictionary learning problem can similarly be formulated as:

$$\text{(DL1)} \quad \begin{cases} \inf\limits_{Z,X,Y} & \iota_S(X) + \|Y\|_1 + \frac{1}{2}\|Z - B\|^2 \\ & Z = XY. \end{cases}$$

Other variants of dictionary learning such as *convolutional dictionary learning* (CDL), that cannot readily be handled by the method in [118], have a biaffine formulation which is nearly identical to (DL1), and can be solved using ADMM. For more information on dictionary learning, see [119, 78, 79, 118, 120].

### 5.3.2 Non-Convex Reformulations of Convex Problems

Recently, various low-rank matrix and tensor recovery problems have been shown to be efficiently solvable by applying first-order methods to nonconvex reformulations of them. For example, the convex *Robust Principal Component Analysis* (RPCA) [121, 122] problem

$$\text{(RPCA1)} \quad \begin{cases} \inf\limits_{L,S} & \|L\|_* + \lambda\|S\|_1 \\ & L + S = B \end{cases}$$

can be reformulated as the biaffine problem

$$\text{(RPCA2)} \quad \begin{cases} \inf\limits_{U,V,S} & \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2) + \lambda\|S\|_1 \\ & UV^T + S = B \\ & U \in \mathbb{R}^{m\times k}, V \in \mathbb{R}^{n\times n}, S \in \mathbb{R}^{m\times n} \end{cases}$$

as long as $k \geq \operatorname{rank}(L^*)$, where $L^*$ is an optimal solution of (RPCA1). See [123] for a proof of this, and applications of the factorization $UV^T$ to other problems. This is also related to the *Burer-Monteiro approach* [124] for semidefinite programming. We remark that (RPCA2) does not satisfy all the assumptions needed for the convergence of ADMM (see A 1.3 and Section 5.4.2), so slack variables must be added.

### 5.3.3 Max-Cut

Given a graph $G = (V, E)$ and edge weights $w \in \mathbb{R}^E$, the (weighted) maximum cut problem is to find a subset $U \subseteq V$ so that $\sum_{u \in U, v \notin U} w_{uv}$ is maximized. This problem is well-known to be NP-hard [125]. An approximation algorithm using semidefinite programming can be shown to achieve an approximation ratio of roughly $0.878$ [126]. Applying the Burer-Monteiro approach [124] to the max-cut semidefinite program [126] with a rank-one constraint, and introducing a slack variable (see A 1.2), we obtain the problem

$$\text{(MC1)} \quad \begin{cases} \sup\limits_{Z,x,y,s} & \frac{1}{2} \sum\limits_{uv \in E} w_{uv}(1 - Z_{uv}) + \frac{\mu_1}{2} \sum\limits_{u \in V} (Z_{uu} - 1)^2 + \frac{\mu_2}{2} \|s\|^2 \\ & Z = xy^T,\ x - y = s. \end{cases}$$

It is easy to verify that all subproblems have very simple closed-form solutions.

### 5.3.4 Risk Parity Portfolio Selection

Given assets indexed by $\{1, \ldots, n\}$, the goal of risk parity portfolio selection is to construct a portfolio weighting $x \in \mathbb{R}^n$ in which every asset contributes an equal amount of risk. This can be formulated with quadratic constraints; see [127] for details. The feasibility problem in [127] is

$$\text{(RP)} \quad \begin{cases} x_i(\Sigma x)_i = x_j(\Sigma x)_j \quad \forall i, j \\ a \leq x \leq b, \quad x_1 + \ldots + x_n = 1 \end{cases}$$

where $\Sigma$ is the (positive semidefinite) *covariance matrix* of the asset returns, and $a$ and $b$ contain lower and upper bounds on the weights, respectively. The authors in [127] introduce a variable $y = x$ and solve (RP) using ADMM by replacing the quadratic risk-parity constraint by a *fourth-order* penalty function $f(x, y, \theta) = \sum_{i=1}^n (x_i(\Sigma y)_i - \theta)^2$. To rewrite this problem with a bilinear constraint, let $\circ$ denote the Hadamard product $(x \circ y)_i = x_i y_i$ and let $P$ be the matrix $\begin{pmatrix} 0 & 0 \\ e_{n-1} & -I_{n-1} \end{pmatrix}$, where $e_n$ is the all-ones vector of length $n$. Let $X$ be the set of permissible

portfolio weights $X = \{x \in \mathbb{R}^n : a \leq x \leq b\} \cap \{x \in \mathbb{R}^n : e_n^T x = 1\}$, and let $\iota_X$ be its indicator function. Then we obtain the problem

$$\text{(RP1)} \quad \begin{cases} \inf\limits_{x,y,z,s} & \iota_X(x) + \frac{\mu}{2}(\|z\|^2 + \|s\|^2) \\ & P(x \circ y) = z \\ & y - \Sigma x = s \end{cases}$$

where we have introduced a slack variable $s$ (see A 1.2).

### 5.3.5 Training Neural Networks

An alternating minimization approach is proposed in [114] for training deep neural networks. By decoupling the linear and nonlinear elements of the network, the backpropagation required to compute the gradient of the network is replaced by a series of subproblems which are easy to solve and readily parallelized. For a network with $L$ layers, let $X_\ell$ be the matrix of edge weights for $1 \leq \ell \leq L$, and let $a_\ell$ be the output of the $\ell$-th layer for $0 \leq \ell \leq L-1$. Deep neural networks are defined by the structure $a_\ell = h(X_\ell a_{\ell-1})$, where $h(\cdot)$ is an *activation function*, which is often taken to be the rectified linear unit (ReLU) $h(z) = \max\{z, 0\}$. The splitting used in [114] introduces new variables $z_\ell$ for $1 \leq \ell \leq L$ so that the network layers are no longer directly connected, but are instead coupled through the relations $z_\ell = X_\ell a_{\ell-1}$ and $a_\ell = h(z_\ell)$.

Let $E(\cdot, \cdot)$ be an error function, and $R$ a regularization function on the weights. Given a matrix of labeled training data $(a_0, y)$, the learning problem is

$$\text{(DNN1)} \quad \begin{cases} \inf\limits_{\{X_\ell\},\{a_\ell\},\{z_\ell\}} & E(z_L, y) + R(X_1, \ldots, X_L) \\ & z_\ell - X_\ell a_{\ell-1} = 0 \text{ for } 1 \leq \ell \leq L \\ & a_\ell - h(z_\ell) = 0 \text{ for } 1 \leq \ell \leq L-1. \end{cases}$$

The algorithm proposed in [114] does not include any regularization $R(\cdot)$, and replaces *both* sets of constraints by quadratic penalty terms in the objective, while maintaining Lagrange multi-

pliers only for the final constraint $z_L = W_L a_{L-1}$. However, since all of the equations $z_\ell = X_\ell a_{\ell-1}$ are biaffine, we can include them in a biaffine formulation of the problem:

$$\text{(DNN2)} \quad \begin{cases} \inf\limits_{\{X_\ell\},\{a_\ell\},\{z_\ell\}} & E(z_L, y) + R(X_1, \ldots, X_L) + \frac{\mu}{2} \sum\limits_{\ell=1}^{L-1} (a_\ell - h(z_\ell))^2 \\ & z_\ell - X_\ell a_{\ell-1} = 0 \text{ for } 1 \leq \ell \leq L. \end{cases}$$

To adhere to our convergence theory, it would be necessary to apply smoothing (such as Nesterov's technique [128]) when $h(z)$ is nonsmooth, as is the ReLU. Alternatively, the ReLU can be replaced by an approximation using nonnegativity constraints (see Section 5.8). In practice [114, §7], using the ReLU directly yields simple closed-form solutions, and appears to perform well experimentally. However, no proof of the convergence of the algorithm in [114] is provided.

### 5.4 Main Results

In this section, we state our assumptions and main results. We will show that ADMM (Algorithm 5) applied to solve a multiaffine constrained problem of the form $(P)$ (refer to page 117) produces a bounded sequence $\{(X^{(k)}, \mathcal{Z}^{(k)})\}_{k=0}^{\infty}$, and that every limit point $(\mathcal{X}^*, \mathcal{Z}^*)$ is a constrained stationary point. While there are fairly general conditions under which $\mathcal{Z}^*$ satisfies first-order optimality conditions (see Assumption 1 and the corresponding discussion in Section 5.4.2 of tightness), the situation with $\mathcal{X}^*$ is more complicated because of the many possible structures of multiaffine maps. Accordingly, we divide the convergence proof into two results. Under one broad set of assumptions, we prove that limit points exist, are feasible, and that $\mathcal{Z}^*$ is a blockwise constrained stationary point for the problem with $\mathcal{X}$ fixed at $\mathcal{X}^*$ (Theorem 5.4.1). Then, we present a set of easily-verifiable conditions under which $(\mathcal{X}^*, \mathcal{Z}^*)$ is also a constrained stationary point (Theorem 5.4.3). If the augmented Lagrangian has additional geometric properties (namely, the Kurdyka-Łojasiewicz property (Section 5.5.5)), then $\{(X^{(k)}, \mathcal{Z}^{(k)})\}_{k=0}^{\infty}$ converges to a single limit point $(\mathcal{X}^*, \mathcal{Z}^*)$ (Theorem 5.4.5).

**Algorithm 5** ADMM

**Input:** $(X_0^{(0)}, \ldots, X_n^{(0)}), (Z_0^{(0)}, Z_1^{(0)}, Z_2^{(0)}), (W_1^{(0)}, W_2^{(0)}), \rho$
**for** $k = 0, 1, 2, \ldots$ **do**
  **for** $i = 0, \ldots, n$ **do**
Compute $X_i^{(k+1)} \in \operatorname{argmin}_{X_i} \mathcal{L}(X_0^{(k+1)}, \ldots, X_{i-1}^{(k+1)}, X_i, X_{i+1}^{(k)}, \ldots, X_n^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$
  **end for**
Compute $\mathcal{Z}^{(k+1)} \in \operatorname{argmin}_{\mathcal{Z}} \mathcal{L}(\mathcal{X}^{(k+1)}, \mathcal{Z}, \mathcal{W}^{(k)})$
$\mathcal{W}^{(k+1)} \leftarrow \mathcal{W}^{(k)} + \rho(A(\mathcal{X}^{(k+1)}, {Z_0}^{(k+1)}) + Q({\mathcal{Z}_>}^{(k+1)}))$
**end for**

### 5.4.1 Assumptions and Main Results

We consider two sets of assumption for our analysis. We provide intuition and further discussion of them in Section 5.4.2. (See Section 5.5 for definitions related to convexity and differentiability.)

**Assumption 1.** *Solving problem $(P)$ (refer to page 117), the following hold.*

**A 1.1.** *For sufficiently large $\rho$, every ADMM subproblem attains its optimal value.*

**A 1.2.** $\operatorname{Im}(Q) \supseteq \operatorname{Im}(A)$.

**A 1.3.** *The following statements regarding the objective function $\phi$ and $Q_2$ hold:*

1. *$\phi$ is coercive on the feasible region $\Omega = \{(\mathcal{X}, \mathcal{Z}) : A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) = 0\}$.*

2. *$\psi(\mathcal{Z})$ can be written in the form*

$$\psi(\mathcal{Z}) = h(Z_0) + g_1(Z_S) + g_2(Z_2)$$

*where*

   *(a) $h$ is proper, convex, and lower semicontinuous.*

   *(b) $Z_S$ represents either $Z_1$ or $(Z_0, Z_1)$ and $g_1$ is $(m_1, M_1)$-strongly convex. That is, either $g_1(Z_1)$ is a strongly convex function of $Z_1$ or $g_1(Z_0, Z_1)$ is a strongly convex function of $(Z_0, Z_1)$.*

*(c) $g_2$ is $M_2$-Lipschitz differentiable.*

*3. $Q_2$ is injective.*

While Assumption 1 may appear to be complicated, it is no stronger than the conditions used in analyzing nonconvex, *linearly*-constrained ADMM. A detailed comparison is given in Section 5.4.2.

Under Assumption 1, Algorithm 5 produces a sequence which has limit points, and every limit point $(\mathcal{X}^{(*)}, \mathcal{Z}^{(*)})$ is feasible with $\mathcal{Z}^{(*)}$ a constrained stationary point for problem $(P)$ with $\mathcal{X}$ fixed to $\mathcal{X}^*$.

**Theorem 5.4.1.** *Suppose that Assumption 1 holds. For sufficiently large $\rho$, the sequence $\{(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})\}_{k=0}^{\infty}$ produced by ADMM is bounded, and therefore has limit points. Every limit point $(\mathcal{X}^*, \mathcal{Z}^*, \mathcal{W}^*)$ satisfies $A(\mathcal{X}^*, Z_0^*) + Q(\mathcal{Z}_>^*) = 0$. There exists a sequence $v^{(k)} \in \partial_{\mathcal{Z}} \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ such that $v^{(k)} \to 0$, and thus*

$$0 \in \partial_{\mathcal{Z}} \psi(\mathcal{Z}^*) + C_{\mathcal{X}^*}^T \mathcal{W}^* \tag{5.4.1}$$

*where $C_{\mathcal{X}^*}$ is the linear map $\mathcal{Z} \mapsto A(\mathcal{X}^*, Z_0) + Q(\mathcal{Z}_>)$ and $C_{\mathcal{X}^*}^T$ is its adjoint. That is, $\mathcal{Z}^*$ is a constrained stationary point for the problem*

$$\min_{\mathcal{Z}} \{\psi(\mathcal{Z}) : A(\mathcal{X}^*, Z_0) + Q(\mathcal{Z}_>) = 0\}.$$

**Remark 5.4.2.** *Let $\sigma := \lambda_{min}(Q_2^T Q_2)$ [2] and $\kappa_1 := \frac{M_1}{m_1}$. One can check that it suffices to choose $\rho$ so that*

$$\frac{\sigma\rho}{2} - \frac{M_2^2}{\sigma\rho} > \frac{M_2}{2} \quad \text{and} \quad \rho > \max\left\{ \frac{2M_1\kappa_1}{\lambda_{++}(Q_1^T Q_1)}, \frac{1}{2}(M_1 + M_2) \max\left\{ \sigma^{-1}, \frac{(1+2\kappa_1)^2}{\lambda_{++}(Q_1^T Q_1)} \right\} \right\}. \tag{5.4.2}$$

Note that Assumption 1 makes very few assumptions about $f(\mathcal{X})$ and the map $A$ as a function of $\mathcal{X}$, other than that $A$ is multiaffine. In Section 5.6, we develop general techniques for prov-

[2] See Section 5.5.4 for the definition of $\lambda_{min}$ and $\lambda_{++}$.

ing that $(\mathcal{X}^*, \mathcal{Z}^*)$ is a constrained stationary point. We now present an easily checkable set of conditions, that ensure that the requirements for those techniques are satisfied.

**Assumption 2.** *Solving problem $(P)$, Assumption 1 and the following hold.*

**A 2.1.** *The function $f(\mathcal{X})$ splits into*

$$f(\mathcal{X}) = F(X_0, \ldots, X_n) + \sum_{i=0}^{n} f_i(X_i)$$

*where $F$ is $M_F$-Lipschitz differentiable, the functions $f_0$, $f_1$, ..., and $f_n$ are proper and lower semicontinuous, and each $f_i$ is continuous on $\mathrm{dom}(f_i)$.*

**A 2.2.** *For each $1 \leq \ell \leq n$,[3] at least one of the following two conditions[4] holds:*

1. *(a) $F(X_0, \ldots, X_n)$ is independent of $X_\ell$.*

   *(b) $f_\ell(X_\ell)$ satisfies a strengthened convexity condition (Definition 5.5.15).*

2. *(a) Viewing $A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) = 0$ as a system of constraints [5], there exists an index $r(\ell)$ such that in the $r(\ell)$-th constraint,*

$$A_{r(\ell)}(\mathcal{X}, Z_0) = R_\ell(X_\ell) + A'_\ell(\mathcal{X}_{\neq \ell}, Z_0)$$

   *for an injective linear map $R_\ell$ and a multiaffine map $A'_\ell$. In other words, the only term in $A_{r(\ell)}$ that involves $X_\ell$ is an injective linear map $R_\ell(X_\ell)$.*

   *(b) $f_\ell$ is either convex or $M_\ell$-Lipschitz differentiable.*

**A 2.3.** *At least one of the following holds for $Z_0$:*

[3] Note that we have deliberately excluded $\ell = 0$. A 2.2 is not required to hold for $X_0$.

[4] That is, either (1a) and (1b) hold, or (2a) and (2b) hold.

[5] As an illustrative example, a problem may be formulated with constraints $X_0 X_1 + Z_1 = 0, X_0 + P_1(X_1) + Z_2 = 0, X_0 X_2 + Z_3 = 0, P_2(X_2) + Z_4 = 0$, where $P_1, P_2$ are injective linear maps. The notation $A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>)$ denotes the concatenation of these equations, which can also be seen naturally as a system of four constraints. In this case, the indices $r(\ell) \in \{1, 2, 3, 4\}$, and A 2.2(2a) is satisfied by the second constraint $X_0 + P_1(X_1) + Z_2 = 0$ for the variables $X_0, X_1$ (i.e. $r(0) = r(1) = 2$ and $R_0 = I, R_1 = P_1$), and by the fourth constraint $P_2(X_2) + Z_4 = 0$ for $X_2$.

1. *$h(Z_0)$ satisfies a strengthened convexity condition (Definition 5.5.15).*

2. *$Z_0 \in Z_S$, so $g_1(Z_S)$ is a strongly convex function of $Z_0$ and $Z_1$.*

3. *Viewing $A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) = 0$ as a system of constraints, there exists an index $r(0)$ such that $A_{r(0)}(\mathcal{X}, Z_0) = R_0(Z_0) + A_0'(\mathcal{X})$ for an injective linear map $R_0$ and multiaffine map $A_0'$.*

With these additional assumptions on $f$ and $A$, we have that every limit point $(\mathcal{X}^*, \mathcal{Z}^*)$ is a constrained stationary point of problem $(P)$.

**Theorem 5.4.3.** *Suppose that Assumption 2 holds (and hence, Assumption 1 and Theorem 5.4.1). Then for sufficiently large $\rho$, there exists a sequence $v^{(k)} \in \partial\mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ with $v^{(k)} \to 0$, and thus every limit point $(\mathcal{X}^*, \mathcal{Z}^*)$ is a constrained stationary point of problem $(P)$. Thus, in addition to* (5.4.1)*, $\mathcal{X}^*$ satisfies, for each $0 \leq i \leq n$,*

$$0 \in \nabla_{X_i} F(\mathcal{X}^*) + \partial_{X_i} f_i(X_i^*) + A^T_{X_i,(\mathcal{X}^*_{\neq i}, Z_0^*)} \mathcal{W}^* \tag{5.4.3}$$

*where $A_{X_i,(\mathcal{X}^*_{\neq i}, Z_0^*)}$ is the $X_i$-linear term of $\mathcal{X} \mapsto A(\mathcal{X}, Z_0)$ evaluated at $(\mathcal{X}^*_{\neq i}, Z_0^*)$ (see Definition 5.5.6) and $A^T_{X_i,(\mathcal{X}^*_{\neq i}, Z_0^*)}$ is its adjoint. That is, for each $0 \leq i \leq n$, $X_i^*$ is a constrained stationary point for the problem*

$$\min_{X_i} \{F(\mathcal{X}^*_{\neq i}, X_i) + f_i(X_i) : A(\mathcal{X}^*_{\neq i}, X_i, Z_0^*) + Q(\mathcal{Z}^*_>) = 0\}.$$

**Remark 5.4.4.** *One can check that it suffices to choose $\rho$ so that, in addition to* (5.4.2)*, we have $\rho > \max\{\lambda_{min}^{-1}(R_\ell^T R_\ell)(\mu_\ell + M_F)\}$, where the maximum is taken over all $\ell$ for which A 2.2(2) holds, and*

$$\mu_\ell = \begin{cases} 0 & \text{if } f_\ell \text{ convex} \\ M_\ell & \text{if } f_\ell \text{ nonconvex, Lipschitz differentiable.} \end{cases}$$

It is well-known that when the augmented Lagrangian has a geometric property known as the *Kurdyka-Łojasiewicz (K-Ł)* property (see Section 5.5.5), which is the case for many optimization

problems that occur in practice, then results such as Theorem 5.4.3 can typically be strengthened because the limit point is unique.

**Theorem 5.4.5.** *Suppose that $\mathcal{L}(\mathcal{X}, \mathcal{Z}, \mathcal{W})$ is a K-Ł function. Suppose that Assumption 2 holds, and furthermore, that A 2.2(2) holds for all $X_0, X_1, \ldots, X_n$[6], and A 2.3(2) holds. Then for sufficiently large $\rho$, the sequence $\{(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})\}_{k=0}^{\infty}$ produced by ADMM converges to a unique constrained stationary point $(\mathcal{X}^*, \mathcal{Z}^*, \mathcal{W}^*)$.*

In Section 5.6, we develop general properties of ADMM that hold without relying on Assumption 1 or Assumption 2. In Section 5.7, the general results are combined with Assumption 1 and then with Assumption 2 to prove Theorem 5.4.1 and Theorem 5.4.3, respectively. Finally, we prove Theorem 5.4.5 assuming that the augmented Lagrangian is a K-Ł function. The results of Section 5.6 may also be useful for analyzing ADMM, since the assumptions required are weak.

### 5.4.2 Discussion of Assumptions

Assumptions 1 and 2 are admittedly long and somewhat involved. In this section, we will discuss them in detail and explore the extent to which they are tight. Again, we wish to emphasize that despite the additional complexity of multiaffine constraints, the basic content of these assumptions is fundamentally the same as in the linear case. There is also a relation between Assumption 2 and proximal ADMM, by which A 2.2(2) can be viewed as introducing a proximal term. This is described in Section 5.4.2.

#### Assumption 1.1

This assumption is necessary for ADMM (Algorithm 5) to be well-defined. We note that this can fail in surprising ways; for instance, the conditions used in [39] are insufficient to guarantee that the ADMM subproblems have solutions. In [129], an example is constructed which satisfies the conditions in [39], and yet the ADMM subproblem fails to attain its (finite) optimal value.

[6]Note that $X_0$ is included here, unlike in Assumption 2.

**Assumption 1.2**

The condition that $\mathrm{Im}(Q) \supseteq \mathrm{Im}(A)$ plays a crucial role at multiple points in our analysis because $\mathcal{Z}_>$, a subset of the *final* block of variables, has a close relation to the dual variables $\mathcal{W}$. It would greatly broaden the scope of ADMM, and simplify modeling, if this condition could be relaxed, but unfortunately this condition is tight for general problems. The following example demonstrates that ADMM is not globally convergent when A 1.2 does not hold, even if the objective function is strongly convex.

**Theorem 5.4.6.** *Consider the problem*

$$\min_{x,y}\{x^2 + y^2 : xy = 1\}.$$

*If the initial point is* $(x^{(0)}, 0, w^{(0)})$*, or if* $w^{(k)} = \rho$ *for some* $k$*, then the ADMM sequence satisfies* $(x^{(k)}, y^{(k)}) \to (0, 0)$ *and* $w^{(k)} \to -\infty$.

*Proof.* The augmented Lagrangian of this problem is $\mathcal{L}(x, y, w) = x^2+y^2+w(xy-1)+\frac{\rho}{2}(xy-1)^2$, and thus $\frac{\partial}{\partial x}\mathcal{L}(x, y, w) = x(2 + \rho y^2) + y(w - \rho)$. If $y = 0$ or $w = \rho$, the minimizer of the $x$-subproblem is $x^+ = 0$. Likewise, if $x = 0$, then $y^+ = 0$. Hence, if either $y^{(k)} = 0$ or $w^{(k)} = \rho$, we have $(x^{(j)}, y^{(j)}) = (0, 0)$ for all $j > k$. The multiplier update is then $w^+ = w - \rho$, so $w^{(k)} \to -\infty$. □

Even for *linearly*-constrained, convex, multiblock problems, this condition[7] is close to indispensable. When all the other assumptions except A 1.2 are satisfied, ADMM can still diverge if $\mathrm{Im}(Q) \not\supseteq \mathrm{Im}(A)$. In fact, [92, Thm 3.1] exhibits a simple 3-block convex problem with objective function $\phi \equiv 0$ on which ADMM diverges for any $\rho$. This condition is used explicitly [109, 110, 112] and implicitly [108] in other analyses of multiblock (nonconvex) ADMM.

[7]For linear constraints $A_1x_1 + \ldots + A_nx_n = b$, the equivalent statement is that $\mathrm{Im}(A_n) \supseteq \bigcup_{i=1}^{n-1} \mathrm{Im}(A_i)$.

**Assumption 1.3**

This assumption posits that the entire objective function $\phi$ is coercive on the feasible region, and imposes several conditions on the term $\psi(\mathcal{Z})$ for the final block $\mathcal{Z}$.

Let us first consider the conditions on $\psi$. The block $\mathcal{Z}$ is composed of three sub-blocks $Z_0, Z_1, Z_2$, and $\psi(\mathcal{Z})$ decomposes as $h(Z_0) + g_1(Z_S) + g_2(Z_2)$, where $Z_S$ represents either $Z_1$ or $(Z_0, Z_1)$. There is a distinction between $Z_0$ and $\mathcal{Z}_> = (Z_1, Z_2)$: namely, $Z_0$ may be coupled with the other variables $\mathcal{X}$ in the nonlinear function $A$, whereas $\mathcal{Z}_>$ appears only in the linear function $Q(\mathcal{Z}_>)$ which satisfies $\mathrm{Im}(Q) \supseteq \mathrm{Im}(A)$.

To understand the purpose of this assumption, consider the following 'abstracted' assumptions, which are implied by A 1.3:

**M1** The objective is Lipschitz differentiable with respect to a 'suitable' subset of $\mathcal{Z}$.

**M2** ADMM yields sufficient decrease [130] when updating $\mathcal{Z}$. That is, for some 'suitable' subset $\widetilde{\mathcal{Z}}$ of $\mathcal{Z}$ and some $\epsilon > 0$, we have $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) \geq \epsilon \|\widetilde{\mathcal{Z}} - \widetilde{\mathcal{Z}}^+\|^2$.

A 'suitable' subset of $\mathcal{Z}$ is one whose associated images in the constraints satisfies A 1.2. By design, our formulation $(P)$ uses the subset $\mathcal{Z}_> = (Z_1, Z_2)$ in this role. **M1** follows from the fact that $g_1, g_2$ are Lipschitz differentiable, and the other conditions in A 1.3 are intended to ensure that **M2** holds. For instance, the strong convexity assumption in A 1.3(2) ensures that **M2** holds with respect to $Z_1$ regardless of the properties of $Q_1$. The concept of sufficient decrease for descent methods is discussed in [130].

To connect this to the classical linearly-constrained problem, observe that an assumption corresponding to **M1** is:

**AL** For the problem $(P1)$ (see page 113), $f_n(x_n)$ is Lipschitz differentiable.

Thus, in this sense $\mathcal{Z}_>$ alone corresponds to the final block in the linearly-constrained case. In the multiaffine setting, we can add a sub-block $Z_0$ to the final block $\mathcal{Z}$, a nonsmooth term $h(Z_0)$ to the

objective function and a coupled constraint $A_1(\mathcal{X}, Z_0)$, but only to a limited extent: the interaction of the final block $\mathcal{Z}$ with these elements is limited to the variables $Z_0$.

As with A 1.2, it would expand the scope of ADMM if **AL**, or the corresponding **M1**, could be relaxed. However, we find that for nonconvex problems, **AL** cannot readily be relaxed even in the linearly-constrained case, where **AL** is a standard assumption [109, 110, 112]. Furthermore, an example is given in [109, 36(a)] of a 2-block problem in which the function $f_2(x_2) = \|x_2\|_1$ is nonsmooth, and it is shown that ADMM diverges for any $\rho$ when initialized at a given point. Thus, we suspect that **AL/M1** is tight for general problems, though it may be possible to prove convergence for specific structured problems not satisfying **M1**.

**M1** often has implications for modeling. When a constraint $C(\mathcal{X}) = 0$ fails to have the required structure, one can introduce a new slack variable $Z$, and replace that constraint by $C(\mathcal{X}) - Z = 0$, and add a term $g(Z)$ to the objective function to penalize $Z$. Because of **M1**, exact penalty functions such as $\lambda\|Z\|_1$ or the indicator function of $\{0\}$ fail to satisfy A 1.3, so this reformulation is not exact. Based on the above discussion, this may be a limitation inherent to ADMM (as opposed to merely an artifact of existing proof techniques).

We turn now to **M2**. Note that **AL** corresponds only to **M1**, which is why A 1.3 is more complicated than **AL**. There are two main sub-assumptions within A 1.3 that ensure **M2**: that $g_1$ is strongly convex in $Z_1$, and the map $Q_2$ is injective. These assumptions are *not* tight[8] since **M2** may hold under alternative hypotheses. On the other hand, we are not aware of other assumptions that are as comparably simple *and* apply with the generality of A 1.3; hence we have chosen to adopt the latter. For example, if we restrict the problem structure by assuming that the sub-block $Z_0$ is not present, then the condition that $g_1$ is strongly convex can be relaxed to the weaker condition that $\nabla^2 g_1(Z_1) + \rho Q_1^T Q_1 \succeq mI$ for $m > 0$. However, even in the absence of A 1.3, one might show that specific problems, or classes of structured problems, satisfy the sufficient decrease property, using the general principles of ADMM outlined in Section 5.6.

Property **M2** often arises implicitly when analyzing ADMM. In some cases, such as [108,

[8] in the sense that this *exact* assumption is always necessary and cannot be replaced.

131], it follows either from strong convexity of the objective function, or because $A_n = I$ (and is thus injective). Proximal and majorized versions of ADMM are considered in [112, 110] and add quadratic terms which force **M2** to be satisfied. The approach in [109], by contrast, takes a different approach and uses an abstract assumption which relates $\|A_k(x_k^+ - x_k)\|^2$ to $\|x_k^+ - x_k\|^2$; in our experience, it is difficult to verify this abstract assumption in general, except when other properties such as strong convexity or injectivity of $A_k$ hold.

Finally, we remark on the coercivity of $\phi$ over the feasible region. It is common to assume coercivity (see, e.g. [110, 109]) to ensure that the sequence of iterates is bounded, which implies that limit points exist. In many applications, such as (DL) (Section 5.3.1), $\phi$ is independent of some of the variables. However, $\phi$ can still be coercive over the feasible region. For the variable-splitting formulation (DL3), this holds because of the constraints $X = X' + X''$ and $Y = Y' + Y''$. The objective function is coercive in $X'$, $X''$, $Y'$, and $Y''$, and therefore $X$ and $Y$ cannot diverge on the feasible region.

**Assumption 2.1**

The key element of this assumption is that $X_0, \ldots, X_n$ may only be coupled by a Lipschitz differentiable function $F(X_0, \ldots, X_n)$, and the (possibly nonsmooth) terms $f_0(X_0), \ldots, f_n(X_n)$ must be separable. This type of assumption is also used in previous works such as [132, 112, 109].

**Assumption 2.2, 2.3**

We have grouped A 2.2, A 2.3 together here because their motivation is the same. Our goal is to obtain conditions under which the convergence of the function differences $\mathcal{L}(\mathcal{X}_{<\ell}^+, X_\ell, \mathcal{X}_{>\ell}, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}_{<\ell}^+, X_\ell^+, \mathcal{X}_{>\ell}, \mathcal{Z}, \mathcal{W})$ implies that $\|X_\ell - X_\ell^+\| \to 0$ (and likewise for $Z_0$). This can be viewed as a much weaker analogue of the sufficient decrease property **M2**. In A 2.2 and A 2.3, we have presented several alternatives under which this holds. Under A 2.2(1) and A 2.3(1), the strengthened

convexity condition (Definition 5.5.15), it is straightforward to show that

$$\mathcal{L}(\mathcal{X}_{<\ell}^{+}, X_\ell, \mathcal{X}_{>\ell}, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}_{<\ell}^{+}, X_\ell^{+}, \mathcal{X}_{>\ell}, \mathcal{Z}, \mathcal{W}) \geq \Delta(\|X_\ell - X_\ell^{+}\|) \tag{5.4.4}$$

(and likewise for $Z_0$), where $\Delta(t)$ is the 0-forcing function arising from strengthened convexity. For A 2.2(2) and A 2.3(2), the inequality (5.4.4) holds with $\Delta(t) = at^2$, which is the sufficient decrease condition of [130]. Note that having $\Delta(t) \in O(t^2)$ is important for proving convergence in the K-Ł setting, hence the additional hypotheses in Theorem 5.4.5.

As with A 1.3, the assumptions in A 2.2 and A 2.3 are not tight, because (5.4.4) may occur under different conditions. We have chosen to use this particular set of assumptions because they are easily verifiable, and fairly general. The general results of Section 5.6 may be useful in analyzing ADMM for structured problems when the particular conditions of A 2.2 are not satisfied.

**Connection with proximal ADMM**

When modeling, one may always ensure that A 2.2(2a) is satisfied for $X_\ell$ by introducing a new variable $Z_3$ and a new constraint $X_\ell = Z_3$. This may appear to be a trivial reformulation of the problem, but it in fact promotes regularity of the ADMM subproblem in the same way as introducing a positive semidefinite proximal term.

Generalizing this trick, let $S$ be positive semidefinite, with square root $S^{1/2}$. Consider the constraint $\sqrt{\frac{2}{\rho}}S^{1/2}(X_\ell - Z_3) = 0$. The term of the augmented Lagrangian induced by this constraint is $\|X_\ell - Z_3\|_S^2$, where $\|\cdot\|_S$ is the seminorm $\|X\|_S^2 = \langle X, SX\rangle$ induced by $S$. To see this, let $W_0$ be the Lagrange multiplier corresponding to this constraint.

**Lemma 5.4.7.** *If $W_3^0$ is initialized to 0, then for all $k \geq 1$, $Z_3^k = X_\ell^k$ and $W_3^k = 0$. Consequently, the constraint $\sqrt{\frac{2}{\rho}}S^{1/2}(X_\ell - Z_3) = 0$ is equivalent to adding a proximal term $\|X_\ell - X_\ell^k\|_S^2$ to the minimization problem for $X_\ell$.*

*Proof.* We proceed by induction. Since $Z_3$ is part of the final block and $W_3^k = 0$, the minimization problem for $Z_3^{k+1}$ is $\min_{Z_3} \|S^{1/2}(Z_3 - X_\ell^{k+1})\|^2$, for which $Z_3^{k+1} = X_\ell^{k+1}$ is an optimal solution.

The update for $W_3^{k+1}$ is then $W_3^{k+1} = \rho(X_\ell^{k+1} - Z_3^{k+1}) = 0$. □

Note that proximal ADMM is often preferable to ADMM in practice [133, 134]. ADMM subproblems, which may have no closed-form solution because of the linear mapping in the quadratic penalty term, can often be transformed into a pure proximal mapping with a closed-form solution, by adding a suitable proximal term. Several applications of this approach are developed in [134]. Furthermore, for proximal ADMM, the conditions on $f_i$ in A 2.2(2b) can be slightly weakened, by modifying Lemma 5.6.9 and Corollary 5.7.3 (see Remark 5.6.11) to account for the proximal term as in [112].

## 5.5 Preliminaries

This section is a collection of definitions, terminology, and technical results which are not specific to ADMM. Proofs of the results in this section can be found in **??**, or in the provided references. The reader may wish to proceed directly to Section 5.6 and return here for details as needed.

### 5.5.1 General Subgradients and First-Order Conditions

In order to unify our treatment of first-order conditions, we use the notion of *general subgradients*, which generalize gradients and subgradients. When $f$ is smooth or convex, the set of general subgradients consists of the ordinary gradient or subgradients, respectively. Moreover, some useful functions that are neither smooth nor convex such as the indicator function of certain nonconvex sets possess general subgradients.

**Definition 5.5.1.** *Let $G$ be a closed and convex set. The* tangent cone $T_G(x)$ *of $G$ at the point $x \in G$ is the set of directions $T_G(x) = \mathrm{cl}(\{y - x : y \in G\})$. The* normal cone $N_G(x)$ *is the set $N_G(x) = \{v : \langle v, y - x\rangle \leq 0 \; \forall y \in G\}$.*

**Definition 5.5.2** ([135], 8.3)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and $x \in \mathrm{dom}(f)$. A vector $v$ is a* regular subgradient *of $f$ at $x$, indicated by $v \in \widehat{\partial} f(x)$, if $f(y) \geq f(x) + \langle v, y - x\rangle + o(\|y - x\|)$ for*

*all* $y \in \mathbb{R}^n$. *A vector* $v$ *is a* general subgradient*, indicated by* $v \in \partial f(x)$*, if there exist sequences* $x_n \to x$ *and* $v_n \to v$ *with* $f(x_n) \to f(x)$ *and* $v_n \in \widehat{\partial} f(x_n)$. *A vector* $v$ *is a* horizon subgradient*, indicated by* $v \in \partial^\infty f(x)$*, if there exist sequences* $x_n \to x$*,* $\lambda_n \to 0$*, and* $v_n \in \widehat{\partial} f(x_n)$ *with* $f(x_n) \to f(x)$ *and* $\lambda_n v_n \to v$.

The properties of the general subgradient can be found in [135, §8].

Under the assumption that the objective function is proper and lower semicontinuous, the ADMM subproblems will satisfy a necessary first-order condition.

**Lemma 5.5.3** ([135], 8.15)**.** *Let* $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ *be proper and lower semicontinuous over a closed set* $G \subseteq \mathbb{R}^n$*. Let* $x \in G$ *be a point at which the following constraint qualification is fulfilled: the set* $\partial^\infty f(x)$ *of horizon subgradients contains no vector* $v \neq 0$ *such that* $-v \in N_G(x)$*. Then, for* $x$ *to be a local optimum of* $f$ *over* $G$*, it is necessary that* $0 \in \partial f(x) + N_G(x)$.

For our purposes, it suffices to note that when $G = \mathbb{R}^n$, the constraint qualification is trivially satisfied because $N_G(x) = \{0\}$. In the context of ADMM, this implies that the solution of each ADMM subproblem satisfies the first-order condition $0 \in \partial \mathcal{L}$.

Problem $(P)$ has nonlinear constraints, and thus it is not guaranteed *a priori* that its minimizers satisfy first-order necessary conditions, unless a constraint qualification holds. However, Assumption 1 implies that the *constant rank constraint qualification* (CRCQ) [136, 137] is satisfied by $(P)$, and minimizers of $(P)$ will therefore satisfy first-order necessary conditions as long as the objective function is suitably regular. This follows immediately from A 1.2 and the following lemma.

**Lemma 5.5.4.** *Let* $C(x, z) = A(x) + Qz$*, where* $A(x)$ *is smooth and* $Q$ *is a linear map with* $\mathrm{Im}(Q) \supseteq \mathrm{Im}(A)$*. Then for any points* $x, z$*, and any vector* $w$*,* $(\nabla C(x, z))^T w = 0$ *if and only if* $Q^T w = 0$.

*Proof.* Observe that $\nabla C(x, z) = \begin{pmatrix} \nabla A(x) & Q \end{pmatrix}$. The condition $\mathrm{Im}(Q) \supseteq \mathrm{Im}(A)$ implies that for every $x$, $\mathrm{Im}(Q) \supseteq \mathrm{Im}(\nabla A(x))$, and thus $\mathrm{Null}(Q^T) \subseteq \mathrm{Null}((\nabla A(x))^T)$. The result follows immediately. □

### 5.5.2 Multiaffine Maps

Every multiaffine map can be expressed as a sum of multilinear maps and a constant. This provides a useful concrete representation.

**Lemma 5.5.5.** *Let $\mathcal{M}(X_1, \ldots, X_n)$ be a multiaffine map. Then, $\mathcal{M}$ can be written in the form $\mathcal{M}(X_1, \ldots, X_n) = B + \sum_{j=1}^{m} \mathcal{M}_j(\mathcal{D}_j)$ where $B$ is a constant, and each $\mathcal{M}_j(\mathcal{D}_j)$ is a multilinear map of a subset $\mathcal{D}_j \subseteq (X_1, \ldots, X_n)$.*

*Proof.* We proceed by induction on $n$. When $n = 1$, a multiaffine map is an affine map, so $\mathcal{M}(X_1) = A(X_1) + B$ as desired. Suppose now that the desired result holds for any multiaffine map of $n-1$ variables. Given a subset $S \subseteq \{1, \ldots, n\}$, let $X_S$ denote the point with $(X_S)_j = X_j$ for $j \in S$, and $(X_S)_j = 0$ for $j \notin S$. That is, the variables not in $S$ are set to 0 in $X_S$. Consider the multiaffine map $\mathcal{N}$ given by

$$\mathcal{N}(X_1, \ldots, X_n) = \mathcal{M}(X_1, \ldots, X_n) + \sum_{|S| \leq n-1} (-1)^{n-|S|} \mathcal{M}(X_S)$$

where the sum runs over all subsets $S \subseteq \{1, \ldots, n\}$ with $|S| \leq n-1$. Since $X_S \mapsto \mathcal{M}(X_S)$ is a multiaffine map of $|S|$ variables, the induction hypothesis implies that $\mathcal{M}(X_S)$ can be written as a sum of multilinear maps. Hence, it suffices to show that $\mathcal{N}(X_1, \ldots, X_n)$ is multilinear, in which case $\mathcal{M}(X_1, \ldots, X_n) = \mathcal{N}(X_1, \ldots, X_n) - \sum_S (-1)^{n-|S|} \mathcal{M}(X_S)$ is a sum of multilinear maps.

We verify the condition of multilinearity. Take $k \in \{1, \ldots, n\}$, and write $U = (X_j : j \neq k)$. Since $\mathcal{M}$ is multiaffine, there exists a linear map $A_U(X_k)$ such that $\mathcal{M}(U, X_k) = A_U(X_k) + \mathcal{M}(U, 0)$. Hence, we can write

$$\mathcal{M}(U, X_k + \lambda Y_k) = \mathcal{M}(U, X_k) + \lambda \mathcal{M}(U, Y_k) - \lambda \mathcal{M}(U, 0). \tag{5.5.1}$$

By the definition of $\mathcal{N}$, $\mathcal{N}(U, X_k + \lambda Y_k)$ is equal to

$$\mathcal{M}(U, X_k + \lambda Y_k) + \sum_{k \in S} (-1)^{n-|S|} \mathcal{M}(U_{S\setminus k}, X_k + \lambda Y_k) + \sum_{k \notin S} (-1)^{n-|S|} \mathcal{M}(U_S, 0)$$

where the sum runs over $S$ with $|S| \leq n-1$. Making the substitution (5.5.1) for every $S$ with $k \in S$, we find that $\mathcal{N}(U, X_k + \lambda Y_k)$ is equal to

$$\begin{aligned} &\mathcal{M}(U, X_k) + \lambda \mathcal{M}(U, Y_k) - \lambda \mathcal{M}(U, 0) \\ &+ \sum_{k \in S} (-1)^{n-|S|} (\mathcal{M}(U_{S\setminus k}, X_k) + \lambda \mathcal{M}(U_{S\setminus k}, Y_k) - \lambda \mathcal{M}(U_{S\setminus k}, 0)) \\ &+ \sum_{k \notin S} (-1)^{n-|S|} \mathcal{M}(U_S, 0) \end{aligned} \tag{5.5.2}$$

Our goal is to show that $\mathcal{N}(U, X_k + \lambda Y_k) = \mathcal{N}(U, X_k) + \lambda \mathcal{N}(U, Y_k)$. Since

$$\mathcal{N}(U, Y_k) = \mathcal{M}(U, Y_k) + \sum_{k \in S} (-1)^{n-|S|} \mathcal{M}(U_{S\setminus k}, Y_k) + \sum_{k \notin S} (-1)^{n-|S|} \mathcal{M}(U_S, 0)$$

we add and subtract $\lambda \sum_{k \notin S} (-1)^{n-|S|} \mathcal{M}(U_S, 0)$ in (5.5.2) to obtain the desired expression $\mathcal{N}(U, X_k) + \lambda \mathcal{N}(U, Y_k)$, minus a residual term

$$\lambda \left( \mathcal{M}(U, 0) + \sum_{k \in S} (-1)^{n-|S|} \mathcal{M}(U_{S\setminus k}, 0) + \sum_{k \notin S} (-1)^{n-|S|} \mathcal{M}(U_S, 0) \right)$$

It suffices to show the term in parentheses is 0.

There is exactly one set $S$ with $k \notin S$ with $|S| = n-1$, and for this set, $U_S = U$. For this $S$, the terms $\mathcal{M}(U, 0)$ and $(-1)^{n-(n-1)} \mathcal{M}(U_S, 0)$ cancel out. The remaining terms are

$$\sum_{k \in S, |S| \leq n-1} (-1)^{n-|S|} \mathcal{M}(U_{S\setminus k}, 0) + \sum_{k \notin S, |S| \leq n-2} (-1)^{n-|S|} \mathcal{M}(U_S, 0) \tag{5.5.3}$$

There is a bijective correspondence between $\{S : k \notin S, |S| \leq n-2\}$ and $\{S : k \in S, |S| \leq n-1\}$

given by $S \leftrightarrow S \cup \{k\}$. Since $|S \cup \{k\}| = |S| + 1$, (5.5.3) becomes

$$\sum_{k \notin S, |S| \leq n-2} ((-1)^{n-|S|-1} + (-1)^{n-|S|}) \mathcal{M}(U_S, 0) = 0$$

which completes the proof. □

Let $\mathcal{M}(X_1, \ldots, X_n, Y)$ be multiaffine, with $Y$ a particular variable of interest, and $X = (X_1, \ldots, X_n)$ the other variables. By Lemma 5.5.5, grouping the multilinear terms $\mathcal{M}_j$ depending on whether $Y$ is one of the arguments of $\mathcal{M}_j$, we have

$$\mathcal{M}(X_1, \ldots, X_n, Y) = B + \sum_{j=1}^{m_1} \mathcal{M}_j(\mathcal{D}_j, Y) + \sum_{j=m_1+1}^{m} \mathcal{M}_j(\mathcal{D}_j) \tag{5.5.4}$$

where each $\mathcal{D}_j \subseteq (X_1, \ldots, X_n)$.

**Definition 5.5.6.** *Let $\mathcal{M}(X_1, \ldots, X_n, Y)$ have the structure* (5.5.4)*. Let $\mathcal{F}_Y$ be the space of functions from $Y \to \mathrm{Im}(\mathcal{M})$. Let $\theta_j : \mathcal{D}_j \to \mathcal{F}_Y$ be the map*[9] *given by $(\theta_j(X))(Y) = \mathcal{M}_j(\mathcal{D}_j, Y)$. Here, we use the notation $\theta_j(X)$ for $\theta_j(\mathcal{D}_j)$, with $\mathcal{D}_j$ taking the values in $X$. Finally, let $\mathcal{M}_{Y,X} = \sum_{j=1}^{m_1} \theta_j(X)$.*

*We call $\mathcal{M}_{Y,X}$ the $Y$-*linear term *of $\mathcal{M}$ (evaluated at $X$).*

To motivate this definition, observe that when $X$ is fixed, the map $Y \mapsto \mathcal{M}(X, Y)$ is *affine*, with the linear component given by $\mathcal{M}_{Y,X}$ and the constant term given by $B_X = B + \sum_{j=m_1+1}^{m} \mathcal{M}_j(\mathcal{D}_j)$. When analyzing the ADMM subproblem in $Y$, a multiaffine constraint $\mathcal{M}(X, Y) = 0$ becomes the *linear* constraint $\mathcal{M}_{Y,X}(Y) = -B_X$.

The definition of multilinearity immediately shows the following.

**Lemma 5.5.7.** *$\theta_j$ is a multilinear map of $\mathcal{D}_j$. For every $X$, $\theta_j(X)$ is a linear map of $Y$, and thus $\mathcal{M}_{Y,X}$ is a linear map of $Y$.*

**Example.** *Consider $\mathcal{M}(X_1, X_2, X_3, X_4) = X_1X_2X_3 + X_2X_3X_4 + X_2 + B = 0$ for square matrices $X_1, X_2, X_3, X_4$. Taking $Y = X_3$ as the variable of focus, and $X = (X_1, X_2, X_4)$, we have*

[9] When $j > m_1$, $\theta_j(X)$ is a constant map of $Y$.

$(\theta_1(X))(Y) = X_1X_2Y$, $(\theta_2(X))(Y) = X_2YX_4$, $(\theta_3(X))(Y) = X_2$, $(\theta_4(X))(Y) = B$, *and thus* $\mathcal{M}_{Y,X}$ *is the linear map* $Y \mapsto X_1X_2Y + X_2YX_4$.

Our general results in Section 5.6 require smooth constraints, which holds for multiaffine maps.

**Lemma 5.5.8.** *Multiaffine maps are smooth, and in particular, biaffine maps are Lipschitz differentiable.*

*Proof.* We prove two auxiliary lemmas, from which Lemma 5.5.8 follows as a corollary.

**Lemma 5.5.9.** *Let* $\mathcal{M}$ *be a multilinear map. There exists a constant* $\sigma_M$ *such that* $\|\mathcal{M}(X_1, \ldots, X_n)\| \leq \sigma_M \prod \|X_i\|$.

*Proof.* We proceed by induction on $n$. When $n = 1$, $\mathcal{M}$ is linear. Suppose it holds for any multilinear map of up to $n-1$ blocks. Given $U = (X_1, \ldots, X_{n-1})$, let $\mathcal{M}_U$ be the linear map $\mathcal{M}_U(X_n) = \mathcal{M}(U, X_n)$, and let $\mathcal{F}$ be the family of linear maps $\mathcal{F} = \{\mathcal{M}_U : \|X_1\| = 1, \ldots, \|X_{n-1}\| = 1\}$. Now, given $X_n$, let $\mathcal{M}_{X_n}$ be the *multilinear* map $\mathcal{M}_{X_n}(U) = \mathcal{M}(U, X_n)$. By induction, there exists some $\sigma_{X_n}$ for $\mathcal{M}_{X_n}$. For every $X_n$, we see that

$$\begin{aligned}\sup_{\mathcal{F}} \|\mathcal{M}_U(X_n)\| &= \sup\{\|\mathcal{M}(X_1, \ldots, X_n)\| : \|X_1\| = 1, \ldots, \|X_{n-1}\| = 1\} \\ &= \sup_{\|X_1\|=1,\ldots,\|X_{n-1}\|=1} \|\mathcal{M}_{X_n}(U)\| \leq \sigma_{X_n} < \infty\end{aligned}$$

Thus, the uniform boundedness principle [138] implies that

$$\sigma_M := \sup_{\mathcal{F}} \|\mathcal{M}_U\|_{op} = \sup_{\|X_1\|=1,\ldots,\|X_n\|=1} \|\mathcal{M}(X_1, \ldots, X_n)\| < \infty$$

Given any $\{X_1, \ldots, X_n\}$, we then have $\|\mathcal{M}(X_1, \ldots, X_n)\| \leq \sigma_M \prod \|X_i\|$. □

**Lemma 5.5.10.** *Let* $\mathcal{M}(X_1, \ldots, X_n)$ *be a multilinear map with* $X = (X_1, \ldots, X_n)$ *and* $X' = (X'_1, \ldots, X'_n)$ *being two points with the property that, for all* $i$, $\|X_i\| \leq d$, $\|X'_i\| \leq d$, *and* $\|X_i - X'_i\| \leq \epsilon$. *Then* $\|\mathcal{M}(X) - \mathcal{M}(X')\| \leq n\sigma_M d^{n-1}\epsilon$, *where* $\sigma_M$ *is from Lemma 5.5.9.*

*Proof.* For each $0 \leq k \leq n$, let $X''_k = (X_1, \ldots, X_k, X'_{k+1}, \ldots, X'_n)$. By Lemma 5.5.9, $\|\mathcal{M}(X''_k) - \mathcal{M}(X''_{k-1})\| = \|\mathcal{M}(X_1, \ldots, X_{k-1}, X_k - X'_k, X'_{k+1}, \ldots, X'_n)\|$ is bounded by $\sigma_M d^{n-1}\|X_k - X'_k\| \leq \sigma_M d^{n-1}\epsilon$. Observe that $\mathcal{M}(X) - \mathcal{M}(X') = \sum_{k=1}^n \mathcal{M}(X''_k) - \mathcal{M}(X''_{k-1})$, and thus we obtain $\|\mathcal{M}(X) - \mathcal{M}(X')\| \leq n\sigma_M d^{n-1}\epsilon$. □

□

### 5.5.3 Smoothness, Convexity, and Coercivity

**Definition 5.5.11.** *A function $g$ is $M$-*Lipschitz differentiable *if $g$ is differentiable and its gradient is Lipschitz continuous with modulus $M$, i.e. $\|\nabla g(x) - \nabla g(y)\| \leq M\|x - y\|$ for all $x, y$.*

*A function $g$ is $(m, M)$-*strongly convex *if $g$ is convex, $M$-Lipschitz differentiable, and satisfies $g(y) \geq g(x) + \langle \nabla g(x), y - x\rangle + \frac{m}{2}\|y - x\|^2$ for all $x$ and $y$. The condition number of $g$ is $\kappa := \frac{M}{m}$.*

**Lemma 5.5.12.** *If $g$ is $M$-Lipschitz differentiable, then $|g(y) - g(x) - \langle \nabla g(x), y - x\rangle| \leq \frac{M}{2}\|y - x\|^2$.*

**Lemma 5.5.13.** *If $g(\cdot, \cdot)$ is $M$-Lipschitz differentiable, then for any fixed $y$, the function $h_y(\cdot) = g(\cdot, y)$ is $M$-Lipschitz differentiable. If $g(\cdot, \cdot)$ is $(m, M)$-strongly convex, then $h_y(\cdot)$ is $(m, M)$-strongly convex.*

**Definition 5.5.14.** *A function $\phi$ is said to be* coercive on the set $\Omega$ *if for every sequence $\{x_k\}_{k=1}^\infty \subseteq \Omega$ with $\|x_k\| \to \infty$, then $\phi(x_k) \to \infty$.*

**Definition 5.5.15.** *A function $\Delta : \mathbb{R} \to \mathbb{R}$ is* 0-forcing *if $\Delta \geq 0$, and any sequence $\{t_k\}$ has $\Delta(t_k) \to 0$ only if $t_k \to 0$. A function $f$ is said to satisfy a* strengthened convexity condition *if there exists a 0-forcing function $\Delta$ such that for any $x, y$, and any $v \in \partial f(x)$, $f$ satisfies*

$$f(y) - f(x) - \langle v, y - x\rangle \geq \Delta(\|y - x\|). \tag{5.5.5}$$

**Remark 5.5.16.** *The strengthened convexity condition is stronger than convexity, but weaker than strong convexity. An example is given by higher-order polynomials of degree $d$ composed with the Euclidean norm, which are not strongly convex for $d > 2$. An example is the function $f(x) = \|x\|_2^3$,*

*which is not strongly convex but does satisfy the strengthened convexity condition. This function appears in the context of the* cubic-regularized Newton method *[139]. We note that ADMM can be applied to solve the nonconvex cubic-regularized Newton subproblem*

$$\min_x \frac{1}{2}x^T Ax + b^T x + \frac{\mu}{3}\|x\|_2^3$$

*by performing the splitting* $\min_{x,y} q(x) + h(y)$ *for* $q(x) = \frac{1}{2}x^T Ax + b^T x$ *and* $h(y) = \frac{\mu}{3}\|x\|_2^3$.

*Since we will subsequently show (Theorem 5.4.1) that the sequence of ADMM iterates is bounded, the strengthened convexity condition can be relaxed. It would be sufficient to assume that for every compact set* $G$*, a 0-forcing function* $\Delta_G$ *exists so that* (5.5.5) *holds with* $\Delta_G$ *whenever* $x, y \in G$.

### 5.5.4 Distances and Translations

**Definition 5.5.17.** *For a symmetric matrix* $S$*, let* $\lambda_{min}(S)$ *be the minimum eigenvalue of* $S$*, and let* $\lambda_{++}(S)$ *be the minimum* positive *eigenvalue of* $S$.

**Lemma 5.5.18.** *Let* $R$ *be a matrix and* $y \in \mathrm{Im}(R)$*. Then* $\|y\|^2 \leq \lambda_{++}^{-1}(R^T R)\|R^T y\|^2$.

*Proof.* Let $y = Rs$, so the desired inequality is $\|R^T Rs\|^2 = (Rs)^T RR^T(Rs) \geq \lambda_{++}(R^T R)\|Rs\|^2$. Since $RR^T$ and $R^T R$ have the same positive eigenvalues, it suffices to show that $Rs$ is orthogonal to $\mathrm{Null}(RR^T)$. This is immediate, since $\mathrm{Null}(RR^T) = \mathrm{Null}(R^T) = \mathrm{Col}(R)^\perp$. □

**Lemma 5.5.19.** *Let* $A$ *be a matrix, and* $b, c \in \mathrm{Im}(A)$*. There exists a constant* $\alpha_A$ *with* $\mathrm{dist}(\{x : Ax = b\}, \{x : Ax = c\}) \leq \alpha_A\|b - c\|$*. Furthermore, we may take* $\alpha_A \leq \sqrt{\lambda_{++}^{-1}(AA^T)}$.

*Proof.* Let $A_r$ be a submatrix of $A$ obtained by taking a maximal linearly independent subset of rows, so $A_r$ has full row rank and $A_r^T(A_r A_r^T)^{-1}$ exists. Let $b_r, c_r$ be the submatrices of $b, c$ having rows corresponding to $A_r$. It is easy to verify that $A(A_r^T(A_r A_r^T)^{-1}b_r) = b$ and $A(A_r^T(A_r A_r^T)^{-1}c_r) = c$. Let $\Delta = b_r - c_r$, and note that $\|\Delta\| \leq \|b - c\|$. Then $\mathrm{dist}(\mathcal{U}_1, \mathcal{U}_2)^2 \leq \langle A_r^T(A_r A_r^T)^{-1}\Delta, A_r^T(A_r A_r^T)^{-1}\Delta\rangle = \langle \Delta, (A_r A_r^T)^{-1}\Delta\rangle \leq \|(A_r A_r^T)^{-1}\|_{op}\|b - c\|^2$. Hence we may take $\alpha = \sqrt{\|(A_r A_r^T)^{-1}\|_{op}}$. □

**Lemma 5.5.20.** *Let $g$ be a $(m, M)$-strongly convex function with condition number $\kappa = \frac{M}{m}$, let $\mathcal{C}$ be a closed and convex set with $\mathcal{C}_1 = a + \mathcal{C}$ and $\mathcal{C}_2 = b + \mathcal{C}$ being two translations of $\mathcal{C}$, let $\delta = \|b - a\|$, and let $x^* = \operatorname{argmin}\{g(x) : x \in \mathcal{C}_1\}$ and $y^* = \operatorname{argmin}\{g(y) : y \in \mathcal{C}_2\}$. Then, $\|x^* - y^*\| \leq (1 + 2\kappa)\delta$.*

*Proof.* Let $d = b - a$, and define $\delta = \|d\|$. Define $x' = y^* - d \in \mathcal{C}_1$, $y' = x^* + d \in \mathcal{C}_2$, and $s = x' - x^* \in T_{\mathcal{C}_1}(x^*)$. Let $\sigma = g(y') - g(x^*)$. We can express $\sigma$ as $\sigma = \int_0^1 \nabla g(x^* + td)^T d \, dt$. Since $\nabla g$ is Lipschitz continuous with constant $M$, we have

$$\begin{aligned} g(y^*) - g(x') &= \int_0^1 \nabla g(x' + td)^T d \, dt \\ &= \sigma + \int_0^1 (\nabla g(x' + td) - \nabla g(x^* + td))^T d \, dt \end{aligned}$$

and thus $|g(y^*) - g(x') - \sigma| \leq \int_0^1 \|\nabla g(x' + td) - \nabla g(x^* + td)\|\|d\| \, dt \leq M\|s\|\delta$, by Lipschitz continuity of $\nabla g$. Therefore $g(y^*) \geq g(x') + \sigma - M\|s\|\delta$. Since $g$ is differentiable and $\mathcal{C}_1$ is closed and convex, $x^*$ satisfies the first-order condition $\nabla g(x^*) \in -N_{\mathcal{C}_1}(x^*)$. Hence, since $s \in T_{\mathcal{C}_1}(x^*) = N_{\mathcal{C}_1}(x^*)^\circ$, we have $g(x') \geq g(x^*) + \langle \nabla g(x^*), s \rangle + \frac{m}{2}\|s\|^2 \geq g(x^*) + \frac{m}{2}\|s\|^2$. Combining these inequalities, we have $g(y^*) \geq g(x^*) + \sigma + \frac{m}{2}\|s\|^2 - M\|s\|\delta$. Since $y^*$ attains the minimum of $g$ over $\mathcal{C}_2$, $g(y') \geq g(y^*)$. Thus

$$g(y') = g(x^*) + \sigma \geq g(y^*) \geq g(x^*) + \sigma + \frac{m}{2}\|s\|^2 - M\|s\|\delta$$

We deduce that $\frac{m}{2}\|s\|^2 - M\|s\|\delta \leq 0$, so $\|s\| \leq 2\kappa\delta$. Since $y^* - x^* = s + d$, we have $\|x^* - y^*\| \leq \|s\| + \|d\| \leq \delta + 2\kappa\delta = (1 + 2\kappa)\delta$. □

**Lemma 5.5.21.** *Let $h$ be a $(m, M)$-strongly convex function, $A$ a linear map of $x$, and $\mathcal{C}$ a closed and convex set. Let $b_1, b_2 \in \operatorname{Im}(A)$, and consider the sets $\mathcal{U}_1 = \{x : Ax + b_1 \in \mathcal{C}\}$ and $\mathcal{U}_2 = \{x : Ax + b_2 \in \mathcal{C}\}$, which we assume to be nonempty. Let $x^* = \operatorname{argmin}\{h(x) : x \in \mathcal{U}_1\}$ and $y^* = \operatorname{argmin}\{h(y) : y \in \mathcal{U}_2\}$. Then, there exists a constant $\gamma$, depending on $\kappa$ and $A$ but independent of $\mathcal{C}$, such that $\|x^* - y^*\| \leq \gamma\|b_2 - b_1\|$.*

*Proof.* Note that $x \in \mathcal{U}_1$ is equivalent to $Ax \in -b_1 + \mathcal{C}$, and thus $\mathcal{U}_1 = A^{-1}(-b_1 + \mathcal{C})$, where $A^{-1}(S) = \{x : Ax \in S\}$ is the preimage of a set $S$ under $A$. Since $\mathcal{U}_1$ is the preimage of the closed, convex set $-b_1 + \mathcal{C}$ under a linear map, $\mathcal{U}_1$ is closed and convex. Similarly, $\mathcal{U}_2 = A^{-1}(-b_2 + \mathcal{C})$ is closed and convex.

We claim that $\mathcal{U}_1, \mathcal{U}_2$ are translates. Since $b_1, b_2 \in \mathrm{Col}(A)$, we can find $d$ such that $Ad = b_1 - b_2$. Given $x \in \mathcal{U}_1$, $A(x + d) \in -b_2 + \mathcal{C}$, so $x + d \in \mathcal{U}_2$, and thus $\mathcal{U}_1 + d \subseteq \mathcal{U}_2$. Conversely, given $y \in \mathcal{U}_2$, $A(y - d) \in -b_1 + \mathcal{C}$, so $y - d \in \mathcal{U}_1$ and $\mathcal{U}_1 + d \supseteq \mathcal{U}_2$. Hence $\mathcal{U}_2 = \mathcal{U}_1 + d$. Applying Lemma 5.5.20 to $\mathcal{U}_1, \mathcal{U}_2$, we find that $\|x^* - y^*\| \leq (1 + 2\kappa)\|d\|$. We may choose $d$ to be a solution of minimum norm satisfying $Ad = b_1 - b_2$; applying Lemma 5.5.19 to the spaces $\{x : Ax = 0\}$ and $\{x : Ax = b_1 - b_2\}$, we see that $\|d\| \leq \alpha\|b_1 - b_2\|$, where $\alpha$ depends only on $A$. Hence $\|x^* - y^*\| \leq (1 + 2\kappa)\alpha\|b_2 - b_1\|$. □

### 5.5.5 K-Ł Functions

**Definition 5.5.22.** *Let $f$ be proper and lower semicontinuous. The domain $\mathrm{dom}(\partial f)$ of the general subgradient mapping is the set $\{x : \partial f(x) \neq \emptyset\}$.*

**Definition 5.5.23** ([130], 2.4)**.** *A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is said to have the* Kurdyka-Łojasiewicz (K-Ł) *property at $x \in \mathrm{dom}(\partial f)$ if there exist $\eta \in (0, \infty]$, a neighborhood $U$ of $x$, and a continuous concave function $\varphi : [0, \eta) \to \mathbb{R}$ such that:*

1. *$\varphi(0) = 0$*
2. *$\varphi$ is smooth on $(0, \eta)$*
3. *For all $s \in (0, \eta)$, $\varphi'(s) > 0$*
4. *For all $y \in U \cap \{w : f(x) < f(w) < f(x) + \eta\}$, the Kurdyka-Łojasiewicz inequality holds:*

$$\varphi'(f(y) - f(x))\,\mathrm{dist}(0, \partial f(x)) \geq 1$$

*A proper, lower semicontinuous function $f$ that satisfies the K-Ł property at every point of* $\mathrm{dom}(\partial f)$ *is called a* K-Ł *function.*

A large class of K-L functions is provided by the *semialgebraic functions*, which include many functions of importance in optimization.

**Definition 5.5.24** ([130], 2.1)**.** *A subset $S$ of $\mathbb{R}^n$ is* (real) semialgebraic *if there exists a finite number of real polynomial functions $P_{ij}, Q_{ij} : \mathbb{R}^n \to \mathbb{R}$ such that*

$$S = \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{x \in \mathbb{R}^n : P_{ij}(x) = 0, Q_{ij}(x) < 0\}.$$

*A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is* semialgebraic *if its graph $\{(x, y) \in \mathbb{R}^{n+m} : f(x) = y\}$ is a real semialgebraic subset of $\mathbb{R}^{n+m}$.*

The set of semialgebraic functions is closed under taking finite sums and products, scalar products, and composition. The indicator function of a semialgebraic set is a semialgebraic function, as is the generalized inverse of a semialgebraic function. More examples can be found in [140].

The key property of K-Ł functions is that if a sequence $\{x^k\}_{k=0}^{\infty}$ is a 'descent sequence' with respect to a K-Ł function, then limit points of $\{x^k\}$ are necessarily unique. This is formalized by the following;

**Theorem 5.5.25** ([130], 2.9)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a proper and lower semicontinuous function. Consider a sequence $\{x^k\}_{k=0}^{\infty}$ satisfying the properties:*

**H1** *There exists $a > 0$ such that for each $k$, $f(x^{k+1}) - f(x^k) \leq -a\|x^{k+1} - x^k\|^2$.*

**H2** *There exists $b > 0$ such that for each $k$, there exists $w^{k+1} \in \partial f(x^{k+1})$ with $\|w^{k+1}\| \leq b\|x^{k+1} - x^k\|$.*

*If $f$ is a K-Ł function, and $x^*$ is a limit point of $\{x^k\}$ with $f(x^k) \to f(x^*)$, then $x^k \to x^*$.*

## 5.6 General Properties of ADMM

In this section, we will derive results that are inherent properties of ADMM, and require minimal conditions on the structure of the problem. We first work in the most general setting where $C$ in the constraint $C(U_0, \ldots, U_n) = 0$ may be any smooth function, the objective function $f(U_0, \ldots, U_n)$ is proper and lower semicontinuous, and the variables $\{U_0, \ldots, U_n\}$ may be coupled. We then specialize to the case where the constraint $C(U_0, \ldots, U_n)$ is multiaffine, which allows us to quantify the changes in the augmented Lagrangian using the subgradients of $f$. Finally, we specialize to the case where the objective function splits into $F(U_0, \ldots, U_n) + \sum_{i=0}^{n} g_i(U_i)$ for a smooth coupling function $F$, which allows finer quantification using the subgradients of the augmented Lagrangian.

The results given in this section hold under very weak conditions; hence, these results may be of independent interest, as tools for analyzing ADMM in other settings.

### 5.6.1 General Objective and Constraints

In this section, we consider

$$\begin{cases} \inf\limits_{U_0,\ldots,U_n} & f(U_0, \ldots, U_n) \\ & C(U_0, \ldots, U_n) = 0. \end{cases}$$

The augmented Lagrangian is given by

$$\mathcal{L}(U_0, \ldots, U_n, W) = f(U_0, \ldots, U_n) + \langle W, C(U_0, \ldots, U_n)\rangle + \frac{\rho}{2}\|C(U_0, \ldots, U_n)\|^2$$

and ADMM performs the updates as in Algorithm 4. We assume only the following.

**Assumption 3.** *The following hold.*

**A 3.1.** *For sufficiently large $\rho$, every ADMM subproblem attains its optimal value.*

**A 3.2.** $C(U_0, \ldots, U_n)$ *is smooth.*

**A 3.3.** *$f(U_0, \ldots, U_n)$ is proper and lower semicontinuous.*

This assumption ensures that the $\operatorname{argmin}$ in Algorithm 4 is well-defined, and that the first-order condition in Lemma 5.5.3 holds at the optimal point. The results in this section are extensions of similar results for ADMM in the classical setting (linear constraints, separable objective function), so it is interesting that the ADMM algorithm retains many of the same properties under the generality of Assumption 3.

**Lemma 5.6.1.** *Let $\mathcal{U} = (U_0, \ldots, U_n)$ denote the set of all variables. The ADMM update of the dual variable $W$ increases the augmented Lagrangian such that $\mathcal{L}(\mathcal{U}^+, W^+) - \mathcal{L}(\mathcal{U}^+, W) = \rho\|C(\mathcal{U}^+)\|^2 = \frac{1}{\rho}\|W - W^+\|^2$. If $\|W - W^+\| \to 0$, then $\nabla_W \mathcal{L}(\mathcal{U}^{(k)}, W^{(k)}) \to 0$ and every limit point $\mathcal{U}^*$ of $\{\mathcal{U}^{(k)}\}_{k=0}^{\infty}$ satisfies $C(\mathcal{U}^*) = 0$.*

*Proof.* The dual update is given by $W^+ = W + \rho C(\mathcal{U}^+)$. Thus, we have

$$\mathcal{L}(\mathcal{U}^+, W^+) - \mathcal{L}(\mathcal{U}^+, W) = \langle W^+ - W, C(\mathcal{U}^+)\rangle = \rho\|C(\mathcal{U}^+)\|^2 = \frac{1}{\rho}\|W - W^+\|^2.$$

For the second statement, observe that $\nabla_W \mathcal{L}(\mathcal{U}, W) = C(\mathcal{U})$. From the dual update, we have $W^+ - W = \rho C(U^+)$. Hence $\|C(\mathcal{U}^+)\| = \frac{1}{\rho}\|W - W^+\| \to 0$. It follows that $\nabla_W \mathcal{L}(U^{(k)}, W^{(k)}) \to 0$ and, by continuity of $C$, any limit point $\mathcal{U}^*$ of $\{U^{(k)}\}_{k=0}^{\infty}$ satisfies $C(\mathcal{U}^*) = 0$. □

Consider the ADMM update of the primal variables. ADMM minimizes $\mathcal{L}(U_0, \ldots, U_n, W)$ with respect to each of the variables $U_0, \ldots, U_n$ in succession. Let $Y = U_j$ be a particular variable of focus, and let $U = \mathcal{U}_{\neq j} = (U_i : i \neq j)$ denote the other variables. For fixed $U$, let $f_U(Y) = f(U, Y)$. When $Y$ is given, we let $U_<$ denote the variables that are updated before $Y$, and $U_>$ the variables that are updated after $Y$. The ADMM subproblem for $Y$ is

$$\min_Y \mathcal{L}(U, Y, W) = \min_Y f_U(Y) + \langle W, C(U, Y)\rangle + \frac{\rho}{2}\|C(U, Y)\|^2.$$

**Lemma 5.6.2.** *The general subgradient of $\mathcal{L}(U, Y, W)$ with respect to $Y$ is given by*

$$\partial_Y \mathcal{L}(U, Y, W) = \partial f_U(Y) + (\nabla_Y C(U, Y))^T W + \rho(\nabla_Y C(U, Y))^T C(U, Y)$$

*where $\nabla_Y C(U, Y)$ is the Jacobian of $Y \mapsto C(U, Y)$ and $(\nabla_Y C(U, Y))^T$ is its adjoint.*

*Defining $V(U, Y, W) = (\nabla_Y C(U, Y))^T W + \rho(\nabla_Y C(U, Y))^T C(U, Y)$, the function $V(U, Y, W)$ is continuous, and $\partial_Y \mathcal{L}(U, Y, W) = \partial f_U(Y) + V(U, Y, W)$. The first-order condition satisfied by $Y^+$ is therefore*

$$\begin{aligned} 0 &\in \partial f_{U^+_<, U_>}(Y^+) + (\nabla_Y C(U^+_<, Y^+, U_>))^T W + \rho(\nabla_Y C(U^+_<, Y^+, U_>))^T C(U^+_<, Y^+, U_>) \\ &= \partial f_{U^+_<, U_>}(Y^+) + V(U^+_<, Y^+, U_>, W). \end{aligned}$$

*Proof.* Since $\langle W, C(U, Y)\rangle + \frac{\rho}{2}\|C(U, Y)\|^2$ is smooth, [135, 8.8(c)] implies that

$$\begin{aligned} \partial_Y \mathcal{L}(U, Y, W) &= \partial f_U(Y) + \nabla_Y \langle W, C(U, Y)\rangle + \nabla_Y \left(\frac{\rho}{2}\|C(U, Y)\|^2\right) \\ &= \partial f_U(Y) + (\nabla_Y C(U, Y))^T W + \rho(\nabla_Y C(U, Y))^T C(U, Y). \end{aligned}$$

□

For the next results, we add the following assumption.

**Assumption 4.** *The function $f$ has the form $f(U_0, \ldots, U_n) = F(U_0, \ldots, U_n) + \sum_{i=0}^n g_i(U_i)$, where $F$ is smooth and each $g_i$ is continuous on* $\mathrm{dom}(g_i)$.

**Lemma 5.6.3.** *Suppose that Assumptions 3 and 4 hold. The general subgradient $\partial_Y \mathcal{L}(U^{(k+1)}, Y^{(k+1)}, W^{(k+1)})$ contains*

$$\begin{aligned} &V(U^{(k+1)}_<, Y^{(k+1)}, U^{(k+1)}_>, W^{(k+1)}) - V(U^{(k+1)}_<, Y^{(k+1)}, U^{(k)}_>, W^{(k)}) \\ &\quad + \nabla_Y F(U^{(k+1)}_<, Y^{(k+1)}, U^{(k+1)}_>) - \nabla_Y F(U^{(k+1)}_<, Y^{(k+1)}, U^{(k)}_>). \end{aligned}$$

*Proof.* Let $g_y$ denote the separable term in $Y$ (that is, if $Y = U_j$, then $g_y = g_j$). By Lemma 5.6.2,

$$\begin{aligned} 0 &\in \partial f_{U_<^{k+1}, U_>^k}(Y^{k+1}) + V(U_<^{k+1}, Y^{k+1}, U_>^k, W^k) \\ &= \nabla_Y F(U_<^{(k+1)}, Y^{(k+1)}, U_>^{(k)}) + \partial g_y(Y^{(k+1)}) + V(U_<^{k+1}, Y^{k+1}, U_>^k, W^k). \end{aligned}$$

Hence,

$$-(\nabla_Y F(U_<^{(k+1)}, Y^{(k+1)}, U_>^{(k)}) + V(U_<^{k+1}, Y^{k+1}, U_>^k, W^k)) \in \partial g_y(Y^{(k+1)}). \tag{5.6.1}$$

In addition, by Lemma 5.6.2,

$$\begin{aligned} &\partial_Y \mathcal{L}(U^{(k+1)}, Y^{(k+1)}, W^{(k+1)}) \\ &= \partial g_y(Y^{(k+1)}) + \nabla_Y F(U_<^{(k+1)}, Y^{(k+1)}, U_>^{(k+1)}) + V(U_<^{(k+1)}, Y^{(k+1)}, U^{(k+1)}, W^{(k+1)}). \end{aligned}$$

Combining this with (5.6.1) implies the desired result.

Applying this to $\partial_Y \mathcal{L}(U^{(k(s))}, Y^{(k(s))}, W^{(k(s))})$, we obtain the subgradient

$$\begin{aligned} v^{(s)} := V(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s))}, W^{(k(s))}) - V(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s)-1)}, W^{(k(s)-1)}) \\ + \nabla_Y F(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s))}) - \nabla_Y F(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s)-1)}). \end{aligned}$$

Since $\{(U^{(k(s))}, Y^{(k(s))}, W^{(k(s))})\}_{s=0}^{\infty}$ converges, and $\|U_>^{(k+1)} - U_>^{(k)}\| \to 0$ and $\|W^{(k+1)} - W^{(k)}\| \to 0$ by assumption, there exists a compact set $\mathcal{B}$ containing the points $\{U_<^{(k(s))}, U_>^{(k(s)-1)}, Y^{(k(s),)}, W^{(k(s))}, W^{(k(s)-1)}\}_s^{\infty}$. $V$ and $\nabla_Y F$ are continuous, so it follows that $V$ and $\nabla_Y F$ are *uniformly* continuous over $\mathcal{B}$. It follows that when $s$ is sufficiently large,

$$V(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s))}, W^{(k(s))}) - V(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s)-1)}, W^{(k(s)-1)})$$

and

$$\nabla_Y F(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s))}) - \nabla_Y F(U_<^{(k(s))}, Y^{(k(s))}, U_>^{(k(s)-1)})$$

can be made arbitrarily small. This completes the proof. □

*Consider any limit point $(U^*, Y^*, W^*)$ of ADMM. If $\|W^+ - W\| \to 0$ and $\|U_>^+ - U_>\| \to 0$, then for any subsequence $\{(U^{(k(s))}, Y^{(k(s))}, W^{(k(s))})\}_{s=0}^{\infty}$ converging to $(U^*, Y^*, W^*)$, there exists a sequence $v^{(s)} \in \partial_Y \mathcal{L}(U^{(k(s))}, Y^{(k(s))}, W^{(k(s))})$ with $v^{(s)} \to 0$.*

**Lemma 5.6.4.** *Suppose that Assumptions 3 and 4 hold. Let $(U^*, Y^*, W^*)$ be a feasible limit point. By passing to a subsequence converging to the limit point, let $\{(U^{(s)}, Y^{(s)}, W^{(s)})\}$ be a subsequence of the ADMM iterates with $(U^{(s)}, Y^{(s)}, W^{(s)}) \to (U^*, Y^*, W^*)$. Suppose that there exists a sequence $\{v^s\}$ such that $v^{(s)} \in \partial_Y \mathcal{L}(U^{(s)}, Y^{(s)}, W^{(s)})$ for all $s$ and $v^{(s)} \to 0$. Then $0 \in \partial g_y(Y^*) + \nabla_Y F(U^*, Y^*) + (\nabla_Y C(U^*, Y^*))^T W^*$, so $(U^*, Y^*, W^*)$ is a constrained stationary point.*

*Proof.* We require the following simple fact.

**Lemma 5.6.5.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. Suppose that we have sequences $x_k \to x$ and $v_k \in \partial f(x_k)$ such that $f(x_k) \to f(x)$ and $v_k \to v$. Then $v \in \partial f(x)$.*

This result would follow by definition if $v_k \in \widehat{\partial} f(x_k)$, but instead we have $v_k \in \partial f(x_k)$. However, for each $k$, there exists sequences $x_{j,k} \to x_k$ and $v_{j,k} \in \widehat{\partial} f(x_{j,k})$ with $f(x_{j,k}) \to f(x_k)$ and $v_{j,k} \to v_k$. By a simple approximation, we can select subsequences $y_s \to x, z_s \in \widehat{\partial} f(y_s)$ with $f(y_s) \to f(x), z_s \to v$. □

*Proof (of Lemma 5.6.4).* By Lemma 5.6.2, $\partial_Y \mathcal{L}(U^{(s)}, Y^{(s)}, W^{(s)}) = \partial g_y(Y^{(s)}) + \nabla_Y F(U^{(s)}, Y^{(s)}) + V(U^{(s)}, Y^{(s)}, W^{(s)})$. Since $V$ is continuous, the sequence $\{V(U^{(s)}, Y^{(s)}, W^{(s)})\}$ converges to $V(U^*, Y^*, W^*)$, which is equal to $(\nabla_Y C(U^*, Y^*))^T W^*$ because $(U^*, Y^*, W^*)$ is feasible. Likewise, $\{\nabla_Y F(U^{(s)}, Y^{(s)})\}$ converges to $\nabla_Y F(U^*, Y^*)$.

Since $v^{(s)} \in \partial_Y \mathcal{L}(U^{(s)}, Y^{(s)}, W^{(s)})$ for all $s$ and $v^{(s)} \to 0$, we deduce that there exists a sequence $\{v_y^{(s)}\}$ such that $v_y^{(s)} \in \partial g_y(Y^{(s)})$ for all $s$ and $v_y^{(s)} \to -(\nabla_Y F(U^*, Y^*) + (\nabla_Y C(U^*, Y^*))^T W^*)$. Hence, by Lemma 5.6.5[10] applied to $g_y$ and the sequences $\{Y^{(s)}\}$ and $\{v_y^{(s)}\}$, we find $-(\nabla_Y F(U^*, Y^*) +$

[10]The assumption that each $g_i$ is continuous on $\mathrm{dom}(g_i)$ was introduced in Assumption 4 to ensure that $g_y(Y^s) \to g_y(Y^*)$, which is required to obtain the general subgradient $\partial g_y(Y^*)$.

$(\nabla_Y C(U^*, Y^*))^T W^*) \in \partial g_y(Y^*)$, as desired. □

**Corollary 5.6.6.** *If Assumptions 3 and 4 hold, and* $\|U_\ell^{(k+1)} - U_\ell^{(k)}\| \to 0$ *for* $\ell \geq 1$*, and* $\|W^{(k+1)} - W^{(k)}\| \to 0$*, then every limit point is a constrained stationary point.*

*Proof.* If $\|W^{(k+1)} - W^{(k)}\| \to 0$ and $\|U_\ell^{(k+1)} - U_\ell^{(k)}\| \to 0$ for all $\ell \geq 1$, then the conditions of Lemma 5.6.3 are satisfied for all blocks $U_0, \ldots, U_n$. Thus, Lemma 5.6.1 implies that $\mathcal{U}^*$ is feasible, and by Lemma 5.6.4, $(\mathcal{U}^*, W^*)$ satisfies the first-order conditions. Note that we do not need to assume $\|U_0^{(k+1)} - U_0^{(k)}\| \to 0$ because $U_0$ is not part of $U_>$ for any block. □

**Remark 5.6.7.** *The assumption that the successive differences* $U_i - U_i^+$ *converge to 0 is used in analyses of nonconvex ADMM such as [141, 142]. Corollary 5.6.6 shows that this is a very strong assumption: it alone implies that every limit point of ADMM is a constrained stationary point, even when* $f$ *and* $C$ *only satisfy Assumptions 3 and 4.*

### 5.6.2 General Objective and Multiaffine Constraints

In this section, we assume that $f$ satisfies Assumption 3 and that $C$ is multiaffine. Note that we do *not* use Assumption 4 in this section.

As in Section 5.6.1, let $Y$ be a particular variable of focus, and $U$ the remaining variables. We let $f_U(Y) = f(U, Y)$. Since $C(U, Y)$ is multiaffine, the resulting function of $Y$ when $U$ is fixed is an *affine* function of $Y$. Therefore, we have $C(U, Y) = C_U(Y) - b_U$ for a *linear* map $C_U$ and a constant $b_U$. The Jacobian of the constraints is then $\nabla_Y C(U, Y) = C_U$ with adjoint $(\nabla_Y C(U, Y))^T = C_U^T$ such that the relation $\langle W, C_U(Y)\rangle = \langle C_U^T W, Y\rangle$ holds.

**Corollary 5.6.8.** *Taking* $\nabla_Y C(U, Y) = C_U$ *in Lemma 5.6.2, the general subgradient of* $Y \mapsto \mathcal{L}(U, Y, W)$ *is given by* $\partial_Y \mathcal{L}(U, Y, W) = \partial f_U(Y) + C_U^T W + \rho C_U^T(C_U(Y) - b_U)$*. Thus, the first-order condition for* $Y \mapsto \mathcal{L}(U, Y, W)$ *at* $Y^+$ *is given by* $0 \in \partial f_U(Y^+) + C_U^T W + \rho C_U^T(C_U(Y^+) - b_U)$.

Using this corollary, we can prove the following.

**Lemma 5.6.9.** *The change in the augmented Lagrangian when the primal variable $Y$ is updated to $Y^+$ is given by*

$$\mathcal{L}(U,Y,W) - \mathcal{L}(U,Y^+,W) = f_U(Y) - f_U(Y^+) - \langle v, Y - Y^+\rangle + \frac{\rho}{2}\|C_U(Y) - C_U(Y^+)\|^2$$

*for some $v \in \partial f_U(Y^+)$.*

*Proof.* Expanding $\mathcal{L}(U,Y,W) - \mathcal{L}(U,Y^+,W)$, the change is equal to

$$\begin{aligned} &f_U(Y) - f_U(Y^+) + \langle W, C_U(Y) - C_U(Y^+)\rangle + \frac{\rho}{2}(\|C_U(Y) - b_U\|^2 - \|C_U(Y^+) - b_U\|^2) \\ &= f_U(Y) - f_U(Y^+) + \langle W, C_U(Y) - C_U(Y^+)\rangle \\ &\quad + \rho\langle C_U(Y) - C_U(Y^+), C_U(Y^+) - b_U\rangle + \frac{\rho}{2}\|C_U(Y) - C_U(Y^+)\|^2. \end{aligned} \tag{5.6.2}$$

To derive (5.6.2), we use the identity $\|Q - P\|^2 - \|R - P\|^2 = \|Q - R\|^2 + 2\langle Q - R, R - P\rangle$ which holds for any elements $P, Q, R$ of an inner product space. Next, observe that

$$\begin{aligned} &\langle W, C_U(Y) - C_U(Y^+)\rangle + \rho\langle C_U(Y) - C_U(Y^+), C_U(Y^+) - b_U\rangle \\ &= \langle C_U(Y) - C_U(Y^+), W + \rho(C_U(Y^+) - b_U)\rangle \\ &= \langle Y - Y^+, C_U^T(W + \rho(C_U(Y^+) - b_U))\rangle \end{aligned}$$

From Corollary 5.6.8, $v = C_U^T W + \rho C_U^T(C_U(Y^+) - b_U)) \in -\partial f_U(Y^+)$. Hence

$$\mathcal{L}(U,Y,W) - \mathcal{L}(U,Y^+,W) = f(Y) - f(Y^+) - \langle v, Y - Y^+\rangle + \frac{\rho}{2}\|C_U(Y) - C_U(Y^+)\|^2.$$

□

**Remark 5.6.10.** *The proof of Lemma 5.6.9 provides a hint as to why ADMM can be extended naturally to multiaffine constraints, but not to arbitrary nonlinear constraints. When $C(U,Y) = 0$ is a general nonlinear system, we cannot manipulate the difference of squares* (5.6.2) *to arrive at the first-order condition for $Y^+$, which uses the crucial fact $\nabla_Y C(U,Y) = C_U$.*

**Remark 5.6.11.** *If we introduce a proximal term $\|Y - Y^k\|_S^2$, the change in the augmented Lagrangian satisfies $\mathcal{L}(U, Y, W) - \mathcal{L}(U, Y^+, W) \geq \|Y - Y^+\|_S^2$, regardless of the properties of $f$ and $C$[11]. This is usually stronger than Lemma 5.6.9. Hence, one can generally obtain convergence of proximal ADMM under weaker assumptions than ADMM.*

Our next lemma shows a useful characterization of $Y^+$.

**Lemma 5.6.12.** *It holds that $Y^+ = \operatorname{argmin}_Y\{f_U(Y) : C_U(Y) = C_U(Y^+)\}$.*

*Proof.* For any two points $Y_1$ and $Y_2$ with $C_U(Y_1) = C_U(Y_2)$, it follows that $\mathcal{L}(U, Y_1, W) - \mathcal{L}(U, Y_2, W) = f_U(Y_1) - f_U(Y_2)$. Hence $Y^+$, the minimizer of $Y \mapsto \mathcal{L}(U, Y, W)$ with $U$ and $W$ fixed, must satisfy $f_U(Y^+) \leq f_U(Y)$ for all $Y$ with $C_U(Y) = C_U(Y^+)$. That is, $Y^+ = \operatorname{argmin}_Y\{f_U(Y) : C_U(Y) = C_U(Y^+)\}$. □

We now show conditions under which the sequence of computed augmented Lagrangian values is bounded below.

**Lemma 5.6.13.** *Suppose that $Y$ represents the final block of primal variables updated in an ADMM iteration and that $f$ is bounded below on the feasible region. Consider the following condition:*

**Condition 5.6.14.** *The following two statements hold true.*

1. *$Y$ can be partitioned*[12] *into sub-blocks $Y = (Y_0, Y_1)$ such that there exists a constant $M_Y$ such that, for any $U$, $Y_0$, $Y_1$, $Y_1'$, and $v \in \partial f_U(Y_0, Y_1)$,*

$$f_U(Y_0, Y_1') - f_U(Y_0, Y_1) - \langle v, (Y_0, Y_1') - (Y_0, Y_1)\rangle \leq \frac{M_Y}{2}\|Y_1' - Y_1\|^2.$$

[11] To see this, define the prox-Lagrangian $\mathcal{L}^P(U, Y, W, O) = \mathcal{L}(U, Y, W) + \|Y - O\|_S^2$. By definition, $Y^+$ decreases the prox-Lagrangian, so $\mathcal{L}^P(U, Y^+, W, Y^k) \leq \mathcal{L}^P(U, Y^k, W, Y^k) = \mathcal{L}(U, Y, W)$ and the desired result follows.

[12] To motivate the sub-blocks $(Y_0, Y_1)$ in Condition 5.6.14, one should look to the decomposition of $\psi(\mathcal{Z})$ in Assumption 1, where we can take $Y_0 = \{Z_0\}$ and $Y_1 = \mathcal{Z}_>$. Intuitively, $Y_1$ is a sub-block such that $\psi$ is a smooth function of $Y_1$, and which is 'absorbing' in the sense that for any $U^+$ and $Y_0^+$, there exists $Y_1$ making the solution feasible.

2. *There exists a constant $\zeta$ such that for every $U^+$ and $Y^+$ produced by ADMM*[13]*, we can find a solution*

$$\widehat{Y}_1 \in \operatorname{argmin}_{Y_1}\{f_{U^+}(Y_0^+, Y_1) : C_{U^+}(Y_0^+, Y_1) = b_{U^+}\}^{14}$$

*satisfying* $\|\widehat{Y}_1 - Y_1^+\|^2 \leq \zeta\|C_{U^+}(Y^+) - b_{U^+}\|^2$.

*If Condition 5.6.14 holds, then there exists $\rho$ sufficiently large such that the sequence $\{\mathcal{L}^{(k)}\}_{k=0}^{\infty}$ is bounded below.*

*Proof.* Suppose that Condition 5.6.14 holds. We proceed to bound the value of $\mathcal{L}^+$ by relating $Y^+$ to the solution $(Y_0^+, \widehat{Y}_1)$. Since $f$ is bounded below on the feasible region and $(U^+, Y_0^+, \widehat{Y}_1)$ is feasible by construction, it follows that $f(U^+, Y_0^+, \widehat{Y}_1) \geq \nu$ for some $\nu > -\infty$. Subtracting $0 = \langle W^+, C_{U^+}(Y_0^+, \widehat{Y}_1) - b_{U^+}\rangle$ from $\mathcal{L}^+$ yields

$$\mathcal{L}^+ = f_{U^+}(Y^+) + \langle W^+, C_{U^+}(Y^+ - (Y_0^+, \widehat{Y}_1))\rangle + \frac{\rho}{2}\|C_{U^+}(Y^+) - b_{U^+}\|^2. \tag{5.6.3}$$

Since $Y$ is the *final* block before updating $W$, all other variables have been updated to $U^+$, and Corollary 5.6.8 implies that the first-order condition satisfied by $Y^+$ is

$$0 \in \partial f_{U^+}(Y^+) + C_{U^+}^T W + \rho C_{U^+}^T(C_{U^+}(Y^+) - b_{U^+}) = \partial f_{U^+}(Y^+) + C_{U^+}^T W^+.$$

Hence $v = C_{U^+}^T W^+ \in -\partial f_{U^+}(Y^+)$. Substituting this into (5.6.3), we have

$$\mathcal{L}^+ = f_{U^+}(Y^+) + \langle v, Y^+ - (Y_0^+, \widehat{Y}_1)\rangle + \frac{\rho}{2}\|C_{U^+}(Y^+) - b_{U^+}\|^2.$$

[13] 2 is assumed to hold for the iterates $U^+$ and $Y^+$ generated by ADMM as the minimal required condition, but one should not, in general, think of this property as being specifically related to the iterates of the algorithm. In the cases we consider, it will be a property of the function $f$ and the constraint $C$ that for *any* point $(\widetilde{U}, \widetilde{Y})$, there exists $\widehat{Y}_1 \in \operatorname{argmin}_{Y_1}\{f_{\widetilde{U}}(\widetilde{Y}_0, Y_1) : C_{\widetilde{U}}(\widetilde{Y}_0, Y_1) = b_{\widetilde{U}}\}$ such that $\|\widehat{Y}_1 - \widetilde{Y}_1\|^2 \leq \zeta\|C_{\widetilde{U}}(Y^+) - b_{\widetilde{U}}\|^2$.

[14] To clarify the definition of $\widehat{Y}_1$, the sub-block for $Y_0$ is fixed to the value of $Y_0^+$ on the given iteration, and then $\widehat{Y}_1$ is obtained by minimizing $f_{U^+}(Y_0^+, Y_1)$ for the $Y_1$ sub-block over the feasible region $C_{U^+}(Y_0^+, Y_1) = b_{U^+}$.

Adding and subtracting $f_{U^+}(Y_0^+, \widehat{Y}_1)$ yields

$$\begin{aligned}\mathcal{L}^+ = f_{U^+}(Y_0^+, \widehat{Y}_1) + \frac{\rho}{2}\|C_{U^+}(Y^+) - b_{U^+}\|^2 \\ - (f_{U^+}(Y_0^+, \widehat{Y}_1) - f_{U^+}(Y^+) - \langle -v, (Y_0^+, \widehat{Y}_1) - Y^+\rangle).\end{aligned}$$

Since $Y^+ = (Y_0^+, Y_1^+)$ and $-v \in \partial f_{U^+}(Y^+)$, Condition 5.6.14 implies that

$$f_{U^+}(Y_0^+, \widehat{Y}_1) - f_{U^+}(Y^+) - \langle -v, (Y_0^+, \widehat{Y}_1) - Y^+\rangle \leq \frac{M_Y}{2}\|\widehat{Y}_1 - Y_1^+\|^2.$$

Hence, we have

$$\begin{aligned}\mathcal{L}^{(+)} &\geq f_{U^+}(Y_0^+, \widehat{Y}_1) + \frac{\rho}{2}\|C_{U^+}(Y^+) - b_{U^+}\|^2 - \frac{M_Y}{2}\|\widehat{Y}_1 - Y_1^+\|^2 \\ &\geq f_{U^+}(Y_0^+, \widehat{Y}_1) + \left(\frac{\rho - M_Y\zeta}{2}\right)\|C_{U^+}(Y^+) - b_{U^+}\|^2. \qquad (5.6.4)\end{aligned}$$

It follows that if $\rho \geq M_Y\zeta$, then $\mathcal{L}^{(k)} \geq \nu$ for all $k \geq 1$. □

The following useful corollary is an immediate consequence of the final inequalities in the proof of the previous lemma.

**Corollary 5.6.15.** *Recall the notation from Lemma 5.6.13. Suppose that $f(U, Y)$ is coercive on the feasible region, Condition 5.6.14 holds, and $\rho$ is chosen sufficiently large so that $\{\mathcal{L}^{(k)}\}$ is bounded above and below. Then $\{U^{(k)}\}$ and $\{Y^{(k)}\}$ are bounded.*

*Proof.* Under the given conditions, $\{\mathcal{L}^{(k)}\}$ is monotonically decreasing and it can be seen from (5.6.4) that $\{f(U^{(k)}, Y_0^{(k)}, \widehat{Y}_1^{(k)})\}$ and $\{\|C_{U^{(k)}}(Y^{(k)}) - b_{U^{(k)}}\|^2\}$ are bounded above. Since $f$ is coercive on the feasible region, and $(U^{(k)}, Y_0^{(k)}, \widehat{Y}_1^{(k)})$ is feasible by construction, this implies that $\{U^{(k)}\}$, $\{Y_0^{(k)}\}$, and $\{\widehat{Y}_1^{(k)}\}$ are bounded. It only remains to show that the 'true' sub-block $\{Y_1^{(k)}\}$ is bounded. From Condition 5.6.14, there exists $\zeta$ with $\|\widehat{Y}_1^{(k)} - Y_1^{(k)}\|^2 \leq \zeta\|C_{U^{(k)}}(Y^{(k)}) - b_{U^{(k)}}\|^2$. (5.6.4) also implies that $\{\|C_{U^{(k)}}(Y^{(k)}) - b_{U^{(k)}}\|^2\}$ is bounded. Hence $\{Y_1^{(k)}\}$ is also bounded. □

### 5.6.3 Separable Objective and Multiaffine Constraints

Now, in addition to Assumption 3, we require that $C(U_0, \ldots, U_n)$ is multiaffine, and that Assumption 4 holds. Most of the results in this section can be obtained from the corresponding results in Section 5.6.1; however, since we will extensively use these results in Section 5.7, it is useful to see their specific form when $C$ is multiaffine.

Again, let $Y = U_j$ be a particular variable of focus, and $U$ the remaining variables. Since $f$ is separable, minimizing $f_U(Y)$ is equivalent to minimizing $f_j(Y)$. Hence, writing $f_y$ for $f_j$, we have

$$\partial_Y \mathcal{L}(U, Y, W) = \partial f_y(Y) + \nabla_Y F(U, Y) + C_U^T W + \rho C_U^T (C_U(Y) - b_U)$$

and $Y^+$ satisfies the first-order condition $0 \in \partial f_y(Y^+) + \nabla_Y F(U, Y^+) + C_U^T W + \rho C_U^T (C_U(Y^+) - b_U)$. The crucial property is that $\partial f_y(Y)$ depends only on $Y$.

**Corollary 5.6.16.** *Suppose that $Y$ is a block of variables in ADMM, and let $U_<, U_>$ be the variables that are updated before and after $Y$, respectively. During an iteration of ADMM, let $C_<(Y) = b_<$ denote the constraint $C(U_<^+, Y, U_>) = b_<$ as a linear function of $Y$, after updating the variables $U_<$, and let $C_>(Y) = b_>$ denote the constraint $C(U_<^+, Y, U_>^+) = b_>$. Then the general subgradient $\partial_Y \mathcal{L}(U_<^+, Y^+, U_>^+, W^+)$ at the final point contains*

$$\begin{aligned}
&(C_>^T - C_<^T) W^+ + C_<^T (W^+ - W) + \rho (C_>^T - C_<^T)(C_>(Y^+) - b_>) \\
&\quad + \rho C_<^T (C_>(Y^+) - b_> - (C_<(Y^+) - b_<)) \\
&\quad + \nabla_Y F(U_<^+, Y^+, U_>^+) - \nabla_Y F(U_<^+, Y^+, U_>)
\end{aligned}$$

*In particular, if $Y$ is the final block, then $C_<^T (W^+ - W) \in \partial_Y \mathcal{L}(U_<^+, Y^+, W^+)$.*

*Proof.* This is an application of Lemma 5.6.3. Since we will use this special case extensively in Section 5.7, we also show the calculation. By Corollary 5.6.8

$$\partial_Y \mathcal{L}(U_<^+, Y^+, U_>^+, W^+) = \partial f_y(Y^+) + \nabla_Y F(U_<^+, Y^+, U_>^+) + C_>^T W^+ + \rho C_>^T (C_>(Y^+) - b_>)$$

By Corollary 5.6.8, $-(\nabla_Y F(U_<^+, Y^+, U_>) + C_<^T W + \rho C_<^T (C_<(Y^+) - b_<)) \in \partial f_y(Y^+)$. To obtain the result, write $C_>^T W^+ - C_<^T W = (C_>^T - C_<^T)W^+ + C_<^T(W^+ - W)$ and

$$C_>^T(C_>(Y^+) - b_>) - C_<^T(C_<(Y^+) - b_<) = (C_>^T - C_<^T)(C_>(Y^+) - b_>) + C_<^T(C_>(Y^+) - b_> - (C_<(Y^+) - b_<)).$$

□

**Lemma 5.6.17.** *Recall the notation from Corollary 5.6.16. Suppose that*

1. $\|W - W^+\| \to 0$,
2. $\|C_> - C_<\| \to 0$,
3. $\|b_> - b_<\| \to 0$, *and*
4. $\{W^{(k)}\}$, $\{Y^{(k)}\}$, $\{C_<^{(k)}\}$, $\{C_>^{(k)}(Y^+) - b_>\}$ *are bounded, and*
5. $\|U_>^+ - U_>\| \to 0$.

*Then there exists a sequence* $v^{(k)} \in \partial_Y \mathcal{L}^{(k)}$ *with* $v^{(k)} \to 0$. *In particular, if* $Y$ *is the* final *block, then only condition 1 and the boundedness of* $\{C_<^{(k)}\}$ *are needed.*

*Proof.* If the given conditions hold, then the triangle inequality and the continuity of $\nabla_Y F$ show that the subgradients identified in Corollary 5.6.16 converge to 0. □

The previous results have focused on a single block $Y$, and the resulting equations $C_U(Y) = b_U$. Let us now relate $C_U, b_U$ to the full constraints. Suppose that we have variables $U_0, \ldots, U_n, Y$ (not necessarily listed in update order), and the constraint $C(U_0, \ldots, U_n, Y) = 0$ is multiaffine. Using the decomposition (5.5.4) and the notation $\theta_j(U)$ from Definition 5.5.6, we express $C_U$ and $b_U$ as

$$C_U = \sum_{j=1}^{m_1} \theta_j(U), \qquad b_U = -(B + \sum_{j=m_1+1}^{m} \theta_j(U)). \tag{5.6.5}$$

This allows us to verify the conditions of Lemma 5.6.17 when certain variables are known to converge.

**Lemma 5.6.18.** *Adopting the notation from Corollary 5.6.16, assume that* $\{U_<^{(k)}\}$, $\{Y^{(k)}\}$, $\{U_>^{(k)}\}$ *are bounded, and that* $\|U_>^+ - U_>\| \to 0$. *Then* $\|C_> - C_<\| \to 0$ *and* $\|b_> - b_<\| \to 0$.

*Proof.* Unpacking our definitions, $C_<$ corresponds to the system of constraints $C(U_<^+, Y, U_>) = b$, and $C_>$ corresponds to $C(U_<^+, Y, U_>^+) = b$. Let $U = (U_<^+, U_>)$ and $U' = (U_<^+, U_>^+)$. By (5.6.5), we have $C_> - C_< = \sum_{j=1}^{m_1} \theta_j(U') - \theta_j(U)$. From Lemma 5.5.8, each $\theta_j$ is smooth, and therefore uniformly continuous over a compact set containing $\{U_<^{(k)}, Y^{(k)}, U_>^{(k)}\}_{k=0}^{\infty}$. Thus, $\|U_>^+ - U_>\| \to 0$ implies that $\|C_> - C_<\| \to 0$. The same applies to $b_> - b_<$. □

## 5.7 Convergence Analysis of Multiaffine ADMM

We now apply the results from Section 5.6 to multiaffine problems of the form $(P)$ that satisfy Assumptions 1 and 2.

### 5.7.1 Proof of Theorem 5.4.1

Under Assumption 1, we prove Theorem 5.4.1. The proof appears at the end of this subsection after we prove a few intermediate results.

**Corollary 5.7.1.** *The general subgradients* $\partial_{\mathcal{Z}} \mathcal{L}(\mathcal{X}, \mathcal{Z}, \mathcal{W})$ *are given by*

$$\partial_{Z_0} \mathcal{L}(\mathcal{X}, Z_0, Z_1, Z_2, \mathcal{W}) = \partial_{Z_0} \psi(\mathcal{Z}) + A_{Z_0,\mathcal{X}}^T \mathcal{W} + \rho A_{Z_0,\mathcal{X}}^T (A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>)) \text{ and}$$
$$\nabla_{Z_i} \mathcal{L}(\mathcal{X}, Z_0, Z_1, Z_2, \mathcal{W}) = \nabla_{Z_i} \psi(\mathcal{Z}) + Q_i^T W_i + \rho Q_i^T (A_i(\mathcal{X}, Z_0) + Q_i(Z_i)) \text{ for } i \in \{1, 2\}.$$

*Proof.* This follows from Corollary 5.6.8. Recall that $A_{Z_0,\mathcal{X}}$ is the $Z_0$-linear term of $Z_0 \mapsto A(\mathcal{X}, Z_0)$ (see Definition 5.5.6). □

**Corollary 5.7.2.** *For all* $k \geq 1$,

$$-\nabla_{Z_i} \psi(\mathcal{Z}^{(k)}) = Q_i^T W_i^{(k)} \quad \text{for } i \in \{1, 2\}. \tag{5.7.1}$$

*Proof.* This follows from Corollaries 5.6.8 and 5.7.1, and the updating formula for $\mathcal{W}^+$ in Algorithm 5. Note that $g_1$ and $g_2$ are smooth, so the first-order conditions for each variable simplifies to

$$-\nabla_{Z_i}\psi(\mathcal{Z}^+) = Q_i^T(W_i + \rho(A_i(\mathcal{X}^+, Z_0^+) + Q_i(Z_i^+))) = Q_i^T W_i^+.$$

Hence, (5.7.1) immediately follows. □

Next, we quantify the decrease in the augmented Lagrangian using properties of $h$, $g_1$, $g_2$, and $Q_2$.

**Corollary 5.7.3.** *The change in the augmented Lagrangian after updating the final block $\mathcal{Z}$ is bounded below by*

$$\frac{m_1}{2}\|Z_S - Z_S^+\|^2 + \left(\frac{\rho\sigma - M_2}{2}\right)\|Z_2 - Z_2^+\|^2, \tag{5.7.2}$$

*where $\sigma = \lambda_{min}(Q_2^T Q_2) > 0$.*

*Proof.* We apply Lemma 5.6.9 to $\mathcal{Z}$. Recall that $\psi = h(Z_0) + g_1(Z_1) + g_2(Z_2)$. The decrease in the augmented Lagrangian is given, for some $v \in \partial h(Z_0^+)$, by

$$\begin{aligned} &h(Z_0) - h(Z_0^+) - \langle v, Z_0 - Z_0^+\rangle + g_1(Z_S) - g_1(Z_S^+) - \langle \nabla g_1(Z_S^+), Z_S - Z_S^+\rangle \\ &+ g_2(Z_2) - g_2(Z_2^+) - \langle \nabla g_2(Z_2^+), Z_2 - Z_2^+\rangle \\ &+ \frac{\rho}{2}\|A_1(\mathcal{X}^+, Z_0 - Z_0^+) + Q_1(Z_1 - Z_1^+)\|^2 + \frac{\rho}{2}\|Q_2(Z_2 - Z_2^+)\|^2. \end{aligned} \tag{5.7.3}$$

By A 1.3, we can show the following bounds for the components of (5.7.3):

1. $h$ is convex, so $h(Z_0) - h(Z_0^+) - \langle v, Z_0 - Z_0^+\rangle \geq 0$.
2. $g_1$ is $(m_1, M_1)$-strongly convex, so $g_1(Z_S) - g_1(Z_S^+) - \langle \nabla g_1(Z_S^+), Z_S - Z_S^+\rangle \geq \frac{m_1}{2}\|Z_S - Z_S^+\|^2$.
3. $g_2$ is $M_2$-Lipschitz differentiable, so $g_2(Z_2) - g_2(Z_2^+) - \langle \nabla g_2(Z_2^+), Z_2 - Z_2^+\rangle \geq -\frac{M_2}{2}\|Z_2 - Z_2^+\|^2$.

Since $Q_2$ is injective, $Q_2^TQ_2$ is positive definite. It follows that with $\sigma = \lambda_{min}(Q_2^TQ_2) > 0$, $\frac{\rho}{2}\|Q_2(Z_2 - Z_2^+)\|^2 \geq \frac{\rho}{2}\sigma\|Z_2 - Z_2^+\|^2$. Since $\frac{\rho}{2}\|A_1(\mathcal{X}^+, Z_0 - Z_0^+) + Q_1(Z_1 - Z_1^+)\|^2 \geq 0$, summing the inequalities establishes the lower bound (5.7.2) on the decrease in $\mathcal{L}$. □

We now bound the change in the Lagrange multipliers by the changes in the variables in $\mathcal{Z}$.

**Lemma 5.7.4.** *We have* $\|\mathcal{W} - \mathcal{W}^+\|^2 \leq \beta_1\|Z_S - Z_S^+\|^2 + \beta_2\|Z_2 - Z_2^+\|^2$, *where* $\beta_1 = M_1^2\lambda_{++}^{-1}(Q_1^TQ_1)$ *and* $\beta_2 = M_2^2\lambda_{++}^{-1}(Q_2^TQ_2) = M_2^2\sigma^{-1}$.

*Proof.* From Corollary 5.7.2, we have $Q_i^TW_i = -\nabla_{Z_i}\psi(\mathcal{Z})$ and $Q_i^TW_i^+ = -\nabla_{Z_i}\psi(\mathcal{Z}^+)$ for $i \in \{1, 2\}$. By definition of the dual update, $W_i^+ - W_i = \rho(A_i(\mathcal{X}^+, Z_0^+) + Q_i(Z_i^+))$. Since $\mathrm{Im}(Q_i)$ contains the image of $A_i$, we have $W_i^+ - W_i \in \mathrm{Im}(Q_i)$. Lemma 5.5.18 applied to $R = Q_i$ and $y = W_i^+ - W_i$ then implies that

$$\|W_i - W_i^+\|^2 \leq \lambda_{++}^{-1}(Q_i^TQ_i)\|Q_i^TW_i - Q_i^TW_i^+\|^2 = \lambda_{++}^{-1}(Q_i^TQ_i)\|\nabla_{Z_i}\psi(\mathcal{Z}) - \nabla_{Z_i}\psi(\mathcal{Z}^+)\|^2.$$

Since $\psi(\mathcal{Z}) = h(Z_0) + g_1(Z_S) + g_2(Z_2)$, we have, for $Z_1$, the bound

$$\begin{aligned}\|\nabla_{Z_1}\psi(\mathcal{Z}) - \nabla_{Z_1}\psi(\mathcal{Z}^+)\|^2 &= \|\nabla_{Z_1}g_1(Z_S) - \nabla_{Z_i}g_1(Z_S^+)\|^2 \\ &\leq \|\nabla g_1(Z_S) - \nabla g_1(Z_S^+)\|^2 \leq M_1^2\|Z_S - Z_S^+\|^2\end{aligned}$$

and thus $\|W_1 - W_1^+\|^2 \leq M_1^2\lambda_{++}^{-1}(Q_1^TQ_1)\|Z_S - Z_S^+\|^2 = \beta_1\|Z_S - Z_S^+\|^2$. A similar calculation applies to $W_2$. Summing over $i \in \{1, 2\}$, we have the desired result. □

**Lemma 5.7.5.** *For sufficiently large* $\rho$, $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}^+) \geq 0$, *and therefore* $\{\mathcal{L}^{(k)}\}_{k=1}^{\infty}$ *is monotonically decreasing. Moreover, for sufficiently small* $\epsilon > 0$, *we may choose* $\rho$ *so that* $\mathcal{L} - \mathcal{L}^+ \geq \epsilon(\|Z_S - Z_S^+\|^2 + \|Z_2 - Z_2^+\|^2)$.

*Proof.* Since the ADMM algorithm involves successively minimizing the augmented Lagrangian over sets of primal variables, it follows that the augmented Lagrangian does not increase after each block of primal variables is updated. In particular, since it does not increase after the update from

$\mathcal{X}$ to $\mathcal{X}^+$, one finds

$$\begin{aligned}\mathcal{L} - \mathcal{L}^+ &= \mathcal{L}(\mathcal{X}, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) + \mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) \\ &\quad + \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}^+) \\ &\geq \mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) + \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}^+).\end{aligned}$$

The only step which increases the augmented Lagrangian is updating $\mathcal{W}$. It suffices to show that the size of $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W})$ exceeds the size of $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}^+)$ by at least $\epsilon(\|Z_S - Z_S^+\|^2 + \|Z_2 - Z_2^+\|^2)$.

By Lemma 5.6.1, $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}^+) = -\frac{1}{\rho}\|\mathcal{W} - \mathcal{W}^+\|^2$. Using Lemma 5.7.4, this is bounded by $-\frac{1}{\rho}(\beta_1\|Z_S - Z_S^+\|^2 + \beta_2\|Z_2 - Z_2^+\|^2)$. On the other hand, eq. equation (5.7.2) of Corollary 5.7.3 implies that $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) \geq \frac{m_1}{2}\|Z_S - Z_S^+\|^2 + \left(\frac{\rho\sigma - M_2}{2}\right)\|Z_2 - Z_2^+\|^2$. Hence, for any $0 < \epsilon < \frac{m_1}{2}$, we may choose $\rho$ sufficiently large so that $\frac{m_1}{2} \geq \frac{\beta_1}{\rho} + \epsilon$ and $\frac{\rho\sigma - M_2}{2} \geq \frac{\beta_2}{\rho} + \epsilon$. □

We next show that $\mathcal{L}^{(k)}$ is bounded below.

**Lemma 5.7.6.** *For sufficiently large $\rho$, the sequence $\{\mathcal{L}^{(k)}\}$ is bounded below, and thus with Lemma 5.7.5, the sequence $\{\mathcal{L}^{(k)}\}$ is convergent.*

*Proof.* We will apply Lemma 5.6.13. By A 1.3, $\phi$ is coercive on the feasible region. Thus, it suffices to show that Condition 5.6.14 holds for the objective function $\phi$ and constraint $A(\mathcal{X}, Z_0) + Q(\mathcal{Z}_>) = 0$, with final block $\mathcal{Z}$.

In the notation of Lemma 5.6.13, we take $Y_0 = \{Z_0\}$, $Y_1 = \mathcal{Z}_> = (Z_1, Z_2)$. We first verify that Condition 5.6.14(1) holds. Recall that $\psi = h(Z_0) + g_1(Z_S) + g_2(Z_2)$ with $g_1$ and $g_2$ Lipschitz

differentiable. Fix any $Z_0$. For any $v \in \partial\psi(Z_0, \mathcal{Z}_>)$, we have

$$\begin{aligned}
&\psi(Z_0, \mathcal{Z}'_>) - \psi(Z_0, \mathcal{Z}_>) - \langle v, (Z_0, \mathcal{Z}'_>) - (Z_0, \mathcal{Z}_>)\rangle \\
&\quad = g_1(Z_0, Z'_1) - g_1(Z_0, Z_1) - \langle \nabla g_1(Z_0, Z_1), (Z_0, Z'_1) - (Z_0, Z_1)\rangle \\
&\qquad + g_2(Z'_2) - g_2(Z_2) - \langle \nabla g_2(Z_2), Z'_2 - Z_2\rangle \\
&\quad \leq \frac{M_1}{2}\|Z'_1 - Z_1\|^2 + \frac{M_2}{2}\|Z'_2 - Z_2\|^2
\end{aligned}$$

Thus, Condition 5.6.14(1) is satisfied with $M_\psi = \frac{1}{2}(M_1 + M_2)$.

Next, we construct $\widehat{\mathcal{Z}_>}$, a minimizer of $\psi(Z_0^+, \mathcal{Z}_>)$ over the feasible region with $\mathcal{X}^+$ and $Z_0^+$ fixed, and find a value of $\zeta$ satisfying Condition 5.6.14(2). There is a unique solution $\widehat{Z}_2$ which is feasible for $A_2(\mathcal{X}^+) + Q_2(Z_2) = 0$, so we take $\widehat{Z}_2 = -Q_2^{-1}A_2(\mathcal{X}^+)$. We find that $\|\widehat{Z}_2 - Z_2^+\|^2 \leq \lambda_{min}^{-1}(Q_2^TQ_2)\|Q_2(Z_2^+ - \widehat{Z}_2)\|^2 = \lambda_{min}^{-1}(Q_2^TQ_2)\|A_2(\mathcal{X}^+) + Q_2(Z_2^+)\|^2$. Thus, if $\zeta \geq \lambda_{min}^{-1}(Q_2^TQ_2)$, then $\|\widehat{Z}_2 - Z_2^+\|^2 \leq \zeta\|A_2(\mathcal{X}^+) + Q_2(Z_2^+)\|^2$.

To construct $\widehat{Z}_1$, consider the spaces $\mathcal{U}_1 = \{Z_1 : Q_1(Z_1) = -A_1(\mathcal{X}^+, Z_0^+)\}$ and $\mathcal{U}_2 = \{Z_1 : Q_1(Z_1) = Q_1(Z_1^+)\}$. From Lemma 5.6.12, $(Z_0^+, Z_1^+)$ is the minimizer of $h(Z_0) + g_1(Z_0, Z_1)$ over the subspace

$$\mathcal{U}_3 = \{(Z_0, Z_1) : A_1(\mathcal{X}^+, Z_0) + Q_1(Z_1) = A_1(\mathcal{X}^+, Z_0^+) + Q_1(Z_1^+)\}.$$

Consider the function $g_0$ given by $g_0(Z_1) = g_1(Z_0^+, Z_1)$. It must be the case that $Z_1^+$ is the minimizer of $g_0$ over $\mathcal{U}_2$, as any other $Z'_1$ with $Q_1(Z'_1) = Q_1(Z_1^+)$ also satisfies $(Z_0^+, Z'_1) \in \mathcal{U}_3$. By Lemma 5.5.13, $g_0$ inherits the $(m_1, M_1)$-strong convexity of $g_1$. Let

$$\widehat{Z}_1 = \text{argmin}_{Z_1}\{g_0(Z_1) : Z_1 \in \mathcal{U}_1\}.$$

Notice that we can express the subspaces $\mathcal{U}_1, \mathcal{U}_2$ as $\mathcal{U}_1 = \{Z_1 | Q_1(Z_1) + A_1(\mathcal{X}^+, Z_0^+) \in \mathcal{C}\}$ and $\mathcal{U}_2 = \{Z_1 | Q_1(Z_1) - Q_1(Z_1^+) \in \mathcal{C}\}$ for the closed convex set $\mathcal{C} = \{0\}$. Since $Z_1^+$ is the minimizer

of $g_0$ over $\mathcal{U}_2$, Lemma 5.5.21 with $h = g_0$, and the subspaces $\mathcal{U}_1$ and $\mathcal{U}_2$, implies that

$$\|\widehat{Z}_1 - Z_1^+\| \leq \gamma \|A_1(\mathcal{X}^+, Z_0^+) + Q_1(Z_1^+)\|$$

where $\gamma$ is dependent only on $\kappa = \frac{M_1}{m_1}$ and $Q_1$. Hence, taking $\zeta = \max\{\gamma^2, \lambda_{min}^{-1}(Q_2^T Q_2)\}$,

$$\begin{aligned}\|\widehat{\mathcal{Z}_>} - \mathcal{Z}_>^+\|^2 &= \|\widehat{Z}_2 - Z_2^+\|^2 + \|\widehat{Z}_1 - Z_1^+\|^2 \\ &\leq \zeta(\|A_2(\mathcal{X}^+) + Q_2(Z_2^+)\|^2 + \|A_1(\mathcal{X}^+, Z_0^+) + Q_1(Z_1^+)\|^2).\end{aligned}$$

Overall, we have shown that Condition 5.6.14 is satisfied. Having verified the conditions of Lemma 5.6.13, we conclude that for sufficiently large $\rho$, $\{\mathcal{L}^{(k)}\}$ is bounded below. □

**Corollary 5.7.7.** *For sufficiently large $\rho$, the sequence $\{(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})\}_{k=0}^{\infty}$ is bounded.*

*Proof.* In Lemma 5.7.6, we showed that Condition 5.6.14 holds. By assumption, $\phi$ is coercive on the feasible region. Thus, the conditions for Corollary 5.6.15 are satisfied, so $\{\mathcal{X}^{(k)}\}$ and $\{\mathcal{Z}^{(k)}\}$ are bounded.

To show that $\{\mathcal{W}^{(k)}\}$ is bounded, recall that $\mathcal{W}^+ - \mathcal{W} \in \mathrm{Im}(Q)$ by A 1.2, and that $Q^T\mathcal{W}^+ = -\nabla_{(Z_1,Z_2)}\psi(\mathcal{Z}^+)$ by Corollary 5.7.2. Taking an orthogonal decomposition of $\mathcal{W}^{(0)}$ for the subspaces $\mathrm{Im}(Q)$ and $\mathrm{Im}(Q)^\perp$, we express $\mathcal{W}^{(0)} = \mathcal{W}_Q^{(0)} + \mathcal{W}_P^{(0)}$, where $\mathcal{W}_Q^{(0)} \in \mathrm{Im}(Q)$ and $\mathcal{W}_P^{(0)} \in \mathrm{Im}(Q)^\perp$. Since $\mathcal{W}^+ - \mathcal{W} \in \mathrm{Im}(Q)$, it follows that if we decompose $\mathcal{W}^{(k)} = \mathcal{W}_Q^{(k)} + \mathcal{W}_P^{(k)}$ with $\mathcal{W}_P^{(k)} \in \mathrm{Im}(Q)^\perp$, then we have $\mathcal{W}_P^{(k)} = \mathcal{W}_P^{(0)}$ for every $k$. Thus, $\|\mathcal{W}^{(k)}\|^2 = \|\mathcal{W}_Q^{(k)}\|^2 + \|\mathcal{W}_P^{(0)}\|^2$ for every $k$. Hence, it suffices to bound $\|\mathcal{W}_Q^{(k)}\|$. Observe that $Q^T\mathcal{W}^{(k)} = Q^T\mathcal{W}_P^{(0)} + Q^T\mathcal{W}_Q^{(k)} = Q^T\mathcal{W}_Q^{(k)}$, because $\mathcal{W}_P^{(0)} \in \mathrm{Im}(Q)^\perp = \mathrm{Null}(Q^T)$. Thus, by Corollary 5.7.2, $Q^T\mathcal{W}_Q^{(k)} = -\nabla_{(Z_1,Z_2)}\psi(\mathcal{Z}^{(k)})$. Since $\{\mathcal{Z}^{(k)}\}$ is bounded and $g_1$ and $g_2$ are Lipschitz differentiable, we deduce that $\{\|Q^T\mathcal{W}_Q^{(k)}\|\}$ is bounded. By Lemma 5.5.18, $\|\mathcal{W}_Q^{(k)}\|^2 \leq \lambda_{++}^{-1}(Q^TQ)\|Q^T\mathcal{W}_Q^{(k)}\|^2$, and so $\{\|\mathcal{W}_Q^{(k)}\|\}$ is bounded. Hence $\{\mathcal{W}^{(k)}\}$ is bounded, completing the proof. □

**Corollary 5.7.8.** *For sufficiently large $\rho$, we have $\|Z_S - Z_S^+\| \to 0$ and $\|Z_2 - Z_2^+\| \to 0$. Consequently, $\|\mathcal{W} - \mathcal{W}^+\| \to 0$ and every limit point is feasible.*

*Proof.* From Lemma 5.7.5, we may choose $\rho$ so that the augmented Lagrangian decreases by at least $\epsilon(\|Z_S - Z_S^+\|^2 + \|Z_2 - Z_2^+\|^2)$ for some $\epsilon > 0$ in each iteration. Summing over $k$, $\epsilon \sum_{k=0}^{\infty} \|Z_S^{(k)} - Z_S^{(k+1)}\|^2 + \|Z_2^{(k)} - Z_2^{(k+1)}\|^2 \leq \mathcal{L}^{(0)} - \lim_k \mathcal{L}^{(k)}$, which is finite by Lemma 5.7.6; hence, $\|Z_S - Z_S^+\| \to 0$ and $\|Z_2 - Z_2^+\| \to 0$.

Using Lemma 5.7.4, $\|\mathcal{W} - \mathcal{W}^+\|^2 \leq \beta_1 \|Z_S - Z_S^+\|^2 + \beta_2 \|Z_2 - Z_2^+\|^2$, so $\|\mathcal{W} - \mathcal{W}^+\| \to 0$ as well. Lemma 5.6.1 then implies that every limit point is feasible. □

Finally, we are prepared to prove the main theorems.

*Proof (of Theorem 5.4.1).* Corollary 5.7.7 implies that limit points of $\{(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})\}$ exist. From Corollary 5.7.8, every limit point is feasible.

We check the conditions of Lemma 5.6.17. Since $\mathcal{Z}$ is the final block, it suffices to verify that $\|\mathcal{W} - \mathcal{W}^+\| \to 0$, and that the maps $\{C_<^{(k)}\}$ are uniformly bounded. That $\|\mathcal{W} - \mathcal{W}^+\| \to 0$ follows from Corollary 5.7.8. Recall from Corollary 5.6.16 that $C_<^{(k)}$ is the $\mathcal{Z}$-linear term of $\mathcal{Z} \mapsto A(\mathcal{X}^{(k)}, Z_0) + Q(\mathcal{Z}_>)$; since $A$ is multiaffine, Lemma 5.5.8 and the boundedness of $\{\mathcal{X}^{(k)}\}_{k=0}^{\infty}$ (Corollary 5.7.7) imply that indeed, $\{C_<^{(k)}\}$ is uniformly bounded in operator norm. Thus, the conditions of Lemma 5.6.17 are satisfied. This exhibits the desired sequence $v^{(k)} \in \partial_{\mathcal{Z}} \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ with $v^{(k)} \to 0$ of Theorem 5.4.1. Lemma 5.6.4 then completes the proof. □

### 5.7.2 Proof of Theorem 5.4.3

Under Assumption 2, we proceed to prove Theorem 5.4.3. For brevity, we introduce the notation $\mathcal{X}_{<i}$ for the variables $(X_0, \ldots, X_{i-1})$ and $\mathcal{X}_{>i}$ for $(X_{i+1}, \ldots, X_n)$.

**Lemma 5.7.9.** *For sufficiently large $\rho$, we have $\|X_\ell - X_\ell^+\| \to 0$ for each $1 \leq \ell \leq n$, and $\|Z_0 - Z_0^+\| \to 0$.*

*Proof.* First, we consider $X_\ell$ for $1 \leq \ell \leq n$. Let $A_X(X_\ell) = b_X$ denote the linear system of constraints when updating $X_\ell$. Recall that under Assumption 2, $f(\mathcal{X}) = F(\mathcal{X}) + \sum_{i=0}^{n} f_i(X_i)$,

where $F$ is a smooth function. By Lemma 5.6.9, the change in the augmented Lagrangian after updating $X_\ell$ is given (for some $v \in \partial f_\ell(X_\ell)$) by

$$f_\ell(X_\ell) - f_\ell(X_\ell^+) - \langle v, X_\ell - X_\ell^+ \rangle + \frac{\rho}{2}\|A_X(X_\ell) - A_X(X_\ell^+)\|^2 \tag{5.7.4}$$
$$+ F(\mathcal{X}_{<\ell}^+, X_\ell, \mathcal{X}_{>\ell}) - F(\mathcal{X}_{<\ell}^+, X_\ell^+, \mathcal{X}_{>\ell}) - \langle \nabla_{X_\ell} F(\mathcal{X}_{<\ell}^+, X_\ell^+, \mathcal{X}_{>\ell}), X_\ell - X_\ell^+ \rangle.$$

By Lemma 5.7.5, the change in the augmented Lagrangian from updating $\mathcal{W}$ is less than the change from updating $\mathcal{Z}$. Since (5.7.4) is nonnegative for every $\ell$, it follows that the change in the augmented Lagrangian in each iteration is greater than the sum of the change from updating each $X_\ell$, and therefore greater than (5.7.4) for each $\ell$. By Lemma 5.7.6, the augmented Lagrangian converges, so the expression (5.7.4) must converge to 0. We will show that this implies the desired result for both cases of A 2.2.

**1** $F(X_0, \ldots, X_n)$ is independent of $X_\ell$ and there exists a 0-forcing function $\Delta_\ell$ such that for any $v \in \partial f_\ell(X_\ell^+)$, $f_\ell(X_\ell) - f_\ell(X_\ell^+) - \langle v, X_\ell - X_\ell^+ \rangle \geq \Delta_\ell(\|X_\ell^+ - X_\ell\|)$. In this case, (5.7.4) is bounded below by $\Delta_\ell(\|X_\ell^+ - X_\ell\|)$. Since (5.7.4) converges to 0, $\Delta_\ell(\|X_\ell^+ - X_\ell\|) \to 0$, which implies that $\|X_\ell - X_\ell^+\| \to 0$.

**2** There exists an index $r(\ell)$ such that $A_{r(\ell)}(\mathcal{X}, Z_0)$ can be decomposed into the sum of a multi-affine map of $\mathcal{X}_{\neq\ell}, Z_0$, and an injective linear map $R_\ell(X_\ell)$. Since $A_X = \nabla_{X_\ell} A(\mathcal{X}, Z_0)$, the $r(\ell)$-th component of $A_X$ is equal to $R_\ell$. Thus, the $r(\ell)$-th component of $A_X(X_\ell) - A_X(X_\ell^+)$ is $R_\ell(X_\ell - X_\ell^+)$.

Let $\mu_\ell = M_\ell$ if $f_\ell$ is $M_\ell$-Lipschitz differentiable, and $\mu_\ell = 0$ if $f_\ell$ is convex and nonsmooth.

We then have

$$\begin{aligned}
&\mathcal{L}(\mathcal{X}^+_{<\ell}, X_\ell, \mathcal{X}_{>\ell}, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+_{<\ell}, X^+_\ell, \mathcal{X}_{>\ell}, \mathcal{Z}, \mathcal{W}) \\
&= f_\ell(X_\ell) - f_\ell(X^+_\ell) - \langle v, X_\ell - X^+_\ell\rangle + \frac{\rho}{2}\|A_X(X_\ell) - A_X(X^+_\ell)\|^2 \\
&\quad + F(\mathcal{X}^+_{<\ell}, X_\ell, \mathcal{X}_{>\ell}) - F(\mathcal{X}^+_{<\ell}, X^+_\ell, \mathcal{X}_{>\ell}) - \langle \nabla_{X_\ell} F(\mathcal{X}^+_{<\ell}, X^+_\ell, \mathcal{X}_{>\ell}), X_\ell - X^+_\ell\rangle \\
&\geq -\frac{(\mu_\ell + M_F)}{2}\|X_\ell - X^+_\ell\|^2 + \frac{\rho}{2}\|R_\ell(X_\ell - X^+_\ell)\|^2 \\
&\geq \frac{1}{2}\left(\rho\lambda_{min}(R_\ell^T R_\ell) - \mu_\ell - M_F\right)\|X_\ell - X^+_\ell\|^2. \qquad (5.7.5)
\end{aligned}$$

Taking $\rho \geq \lambda^{-1}_{min}(R_\ell^T R_\ell)(\mu_\ell + M_F)$, we see that $\|X_\ell - X^+_\ell\| \to 0$.

It remains to show that $\|Z_0 - Z^+_0\| \to 0$ in all three cases of A 2.3. Two cases are immediate. If $Z_0 \in Z_S$, then $\|Z_0 - Z^+_0\| \to 0$ is implied by Corollary 5.7.8, because $\|Z_S - Z^+_S\| \to 0$. If $h(Z_0)$ satisfies a strengthened convexity condition, then by inspecting the terms of equation (5.7.3), we see that the same argument for $X_\ell$ applies to $Z_0$. Thus, we assume that A 2.3(3) holds. Let $A_X(\mathcal{Z}) = b_X$ denote the system of constraints when updating $\mathcal{Z}$. The third condition of A 2.3 implies that for $r = r(0)$, the $r$-th component of the system of constraints $A_1(\mathcal{X}, Z_0) + Q_1(Z_1) = 0$ is equal to $A'_0(\mathcal{X}) + R_0(Z_0) + Q_r(Z_1) = 0$ for the corresponding submatrix $Q_r$ of $Q_1$. Hence, the $r$-th component of $A_X(\mathcal{Z})$ is equal to $R_0(Z_0) + Q_r(Z_1)$. Inspecting the terms of equation (5.7.3), we see that

$$\mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) \geq \frac{\rho}{2}\|R_0(Z_0) + Q_r(Z_1) - (R_0(Z^+_0) + Q_r(Z^+_1))\|^2$$

Since $\mathcal{L}^{(k)}$ converges, and the increases of $\mathcal{L}^{(k)}$ are bounded by $\frac{1}{\rho}\|\mathcal{W} - \mathcal{W}^+\| \to 0$, we must also have $\mathcal{L}(\mathcal{X}^+, \mathcal{Z}, \mathcal{W}) - \mathcal{L}(\mathcal{X}^+, \mathcal{Z}^+, \mathcal{W}) \to 0$, or else the updates of $\mathcal{Z}$ would decrease $\mathcal{L}^{(k)}$ to $-\infty$. By Corollary 5.7.8, $\|Z_1 - Z^+_1\| \to 0$, since $Z_1$ is always part of $Z_S$. Hence $\|R_0(Z_0 - Z^+_0)\| \to 0$, and the injectivity of $R_0$ implies that $\|Z_0 - Z^+_0\| \to 0$. Combined with Corollary 5.7.8, we conclude that $\|\mathcal{Z} - \mathcal{Z}^+\| \to 0$. □

*Proof (of Theorem 5.4.3).* We first confirm that the conditions of Lemma 5.6.17 hold for $\{X_0, \ldots, X_n\}$.

Corollaries 5.7.7 and 5.7.8 together show that all variables and constraints are bounded, and that $\|\mathcal{W} - \mathcal{W}^+\| \to 0$. Since $\|X_\ell - X_\ell^+\| \to 0$ for all $\ell \geq 1$, and $\|\mathcal{Z} - \mathcal{Z}^+\| \to 0$, we have $\|U_>^+ - U_>\| \to 0$, and the conditions $\|C_> - C_<\| \to 0$ and $\|b_> - b_<\| \to 0$ follow from Lemma 5.6.18. Note that $X_0$ is not part of $X_>$ for any $\ell$, which is why we need only that $\{X_0^{(k)}\}$ is bounded, and $\|X_\ell - X_\ell^+\| \to 0$ for $\ell \geq 1$. Thus, Lemma 5.6.17 implies that we can find $v_x^{(k)} \in \partial_{\mathcal{X}} \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ with $v_x^{(k)} \to 0$; combined with the subgradients in $\partial_{\mathcal{Z}} \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ converging to 0 (Theorem 5.4.1) and the fact that $\nabla_W \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)}) \to 0$ (Lemma 5.6.1), we obtain a sequence $v^{(k)} \in \partial \mathcal{L}(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})$ with $v^{(k)} \to 0$.

Having verified the conditions for Lemma 5.6.17, Lemma 5.6.4 then shows that all limit points are constrained stationary points. Part of this theorem (that every limit point is a constrained stationary point) can also be deduced directly from Corollary 5.6.6 and Lemma 5.7.9. □

### 5.7.3 Proof of Theorem 5.4.5

*Proof.* We will apply Theorem 5.5.25 to $\mathcal{L}(\mathcal{X}, \mathcal{Z}, \mathcal{W})$. First, for **H2**, observe that the desired subgradient $w^{k+1}$ is provided by Lemma 5.6.3. Since the functions $V$ and $\nabla F$ are continuous, and all variables are bounded by Corollary 5.7.7, $V$ and $\nabla F$ are *uniformly* continuous on a compact set containing $\{(\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}, \mathcal{W}^{(k)})\}_{k=0}^{\infty}$. Hence, we can find $b$ for which **H2** is satisfied.

Together, Lemma 5.7.4 and Lemma 5.7.5 imply that **H1** holds for $\mathcal{W}$ and $\mathcal{Z}_>$. Using the hypothesis that A 2.2(2) holds for $X_0, X_1, \ldots, X_n$, the inequality (5.7.5) implies that property **H1** in Theorem 5.5.25 holds for $X_0, X_1, \ldots, X_n$. Lastly, A 2.3(2) holds, so $Z_S = (Z_0, Z_1)$ and thus $g_1$ is a strongly convex function of $Z_0$, so Corollary 5.7.3 implies that **H1** also holds for $Z_0$. Thus, we see that **H1** is satisfied for all variables. Finally, Assumption 2 implies that $\phi$, and therefore $\mathcal{L}$, is continuous on its domain, so Theorem 5.5.25 applies and completes the proof. □

## 5.8 Supplementary: Alternate Deep Neural Net Formulation

When $h(z) = \max\{z, 0\}$, we can approximate the constraint $a_\ell - h(z_\ell) = 0$ by introducing a variable $a'_\ell \geq 0$, and minimizing a combination of $\|a'_\ell - z_\ell\|^2$, $\|a'_\ell - a_\ell\|^2$. This leads to the following

biaffine formulation, which satisfies Assumptions 1 and 2, for the deep learning problem:

$$\begin{cases} \inf & E(z_L, y) + \sum_{\ell=1}^{L-1} \iota(a'_\ell) + \frac{\mu}{2} \sum_{\ell=1}^{L-1} [\|\hat{a}_\ell\|^2 + \|s_\ell\|^2] + R(X_1, \ldots, X_L) \\ & X_L a_{L-1} - z_L = 0 \\ & \begin{bmatrix} X_\ell a_{\ell-1} \\ a'_\ell \\ a'_\ell - a_\ell \end{bmatrix} - \begin{bmatrix} I & 0 & 0 \\ I & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} z_\ell \\ s_\ell \\ \hat{a}_\ell \end{bmatrix} = 0 \quad \text{for } 1 \leq \ell \leq L-1. \end{cases} .$$

## 5.9 Supplementary: Formulations with Closed-Form Subproblems

### 5.9.1 Representation Learning

Observe that in (NMF1), the ADMM subproblems for $X$ and $Y$, which have quadratic objective functions and nonnegativity constraints, do not have closed-form solutions. To update $X$ and $Y$, [113] proposes using ADMM to approximately solve the subproblems. This difficulty can be removed through variable splitting. Specifically, by introducing auxiliary variables $X'$ and $Y'$, one obtains the equivalent problem:

$$\text{(NMF2)} \quad \begin{cases} \inf\limits_{X,X',Y,Y',Z} & \iota(X') + \iota(Y') + \frac{1}{2}\|Z - B\|^2 \\ & Z = XY,\ X = X',\ Y = Y', \end{cases}$$

where $\iota$ is the indicator function for the nonnegative orthant; i.e., $\iota(X) = 0$ if $X \geq 0$ and $\iota(X) = \infty$ otherwise. One can now apply ADMM, updating the variables in the order $Y$, $Y'$, $X'$, then $(Z, X)$. Notice that the subproblems for $Y$ and $(Z, X)$ now merely involve minimizing quadratic functions (with no constraints). The solution to the subproblem for $Y'$,

$$\inf_{Y' \geq 0} \langle W, -Y' \rangle + \frac{\rho}{2}\|Y - Y'\|^2 = \inf_{Y' \geq 0} \left\| Y' - (Y + \frac{1}{\rho}W) \right\|^2, \tag{5.9.1}$$

is obtained by setting the negative entries of $Y + \frac{1}{\rho}W$ to 0. An analogous statement holds for $X'$.

Unfortunately, while this splitting and order of variable updates yields easy subproblems, it

does not satisfy all the assumptions we require in A 1.3 (see also Section 5.4.2). One reformulation which keeps all the subproblems easy *and* satisfies our assumptions involves introducing slacks $X''$ and $Y''$ and penalizing them by a smooth function, as in

$$\text{(NMF3)} \quad \begin{cases} \inf\limits_{X,X',X'',Y,Y',Y'',Z} & \iota(X') + \iota(Y') + \frac{1}{2}\|Z - B\|^2 + \frac{\mu}{2}\|X''\|^2 + \frac{\mu}{2}\|Y''\|^2 \\ & Z = XY,\ X = X' + X'',\ Y = Y' + Y''. \end{cases}$$

The variables can be updated in the order $Y$, $Y'$, $X$, $X'$, then $(Z, X'', Y'')$. It is straightforward to verify that the ADMM subproblems either involve minimizing a quadratic (with no constraints) or projecting onto the nonnegative orthant, as in (5.9.1).

Next, we consider (DL). In [118], a block coordinate descent (BCD) method is proposed for solving (DL), which requires an iterative subroutine for the Lasso [143] problem ($L_1$-regularized least squares regression). To obtain easy subproblems, we can formulate (DL) as

$$\text{(DL2)} \quad \begin{cases} \inf\limits_{X,Y,Z,X',Y'} & \iota_S(X') + \|Y'\|_1 + \frac{\mu}{2}\|Z - B\|_2^2 \\ & Z = XY,\ Y = Y',\ X = X'. \end{cases}$$

Notice that the Lasso has been replaced by soft thresholding, which has a closed-form solution. As with (NMF2), not all assumptions in Assumption 1 are satisfied, so to retain easy subproblems and satisfy all assumptions, we introduce slack variables to obtain the problem

$$\text{(DL3)} \quad \begin{cases} \inf\limits_{X,X',X'',Y,Y',Y'',Z} & \iota_S(X') + \|Y'\|_1 + \frac{\mu_Z}{2}\|Z - B\|_2^2 + \frac{\mu_X}{2}\|X''\|^2 + \frac{\mu_Y}{2}\|Y''\|_2^2 \\ & Z = XY,\ Y = Y' + Y'',\ X = X' + X''. \end{cases}$$

### 5.9.2 Risk Parity Portfolio Selection

As before, we can split the variables in a biaffine model to make each subproblem easy to solve. The projection onto the set of permissible weights $X$ has no closed-form solution, so let $X_B$ be the

box $\{x \in \mathbb{R}^n : a \leq x \leq b\}$, and $\iota_{X_B}$ its indicator function. One can then solve:

$$\text{(RP2)} \quad \begin{cases} \inf\limits_{x,x',y,z,z',z'',z'''} & \iota_{X_B}(x') + \frac{\mu}{2}(\|z\|^2 + \|z'\|^2 + \|z''\|^2 + \|z'''\|^2) \\ & P(x \circ y) = z, \quad y = \Sigma x + z' \\ & x = x' + z'', \quad e_n^T x = 1 + z'''. \end{cases}$$

The variables can be updated in the order $x$, $x'$, $y$, $(z, z', z'', z''')$. It is easy to see that every subproblem involves minimizing a quadratic function with no constraints, except for the update of $x'$, which consists of projection onto the box $X_B$ and can be evaluated in closed-form.

# References


[1] W. Gao and D. Goldfarb, "Block BFGS methods," *SIAM Journal on Optimization*, vol. 28, pp. 1205–1231, 2 2018.

[2] ——, "Quasi-newton methods: Superlinear convergence without line searches for self-concordant functions," *Optimization Methods and Software*, vol. 34, no. 1, pp. 194–217, 2018.

[3] Y. Teng, W. Gao, F. Chalus, A. Choromanska, D. Goldfarb, and A. Weller, "Leader stochastic gradient descent for distributed training of deep learning models," *Advances in Neural Information Processing Systems 32*, 2019.

[4] W. Gao, D. Goldfarb, and F. E. Curtis, "ADMM for multiaffine constrained optimization," *Optimization Methods and Software*, vol. 35, no. 2, pp. 257–303, 2020.

[5] A.-L. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Comptes rendus de l'Académie des Sciences Paris*, vol. 25, pp. 536–538, 1847.

[6] F. Viète, "De numerosa potestatum," 1600.

[7] I. Newton, "De analysi per aequationes numero terminorum infinitas," 1669.

[8] ——, *Philosophiae Naturalis Principia Mathematica*. 1687.

[9] J. Raphson, "Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, et expedita, ex nova infinitarum serierum doctrina deducta ac demonstra," 1690.

[10] T. Simpson, "A new treatise of fluxions," 1737.

[11] T. Ypma, "Historical development of the newton-raphson method," *SIAM Review*, vol. 37, no. 4, pp. 531–551, 1995.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019*

*Conference of the North American Chapter of the Association for Computational Linguistics*, vol. 1, 2019, pp. 4171–4186.

[14] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI*, 2019.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Physica-Verlag HD, pp. 177–186.

[17] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[18] M. M. Waldrop, "The chips are down for Moore's law," *Nature News*, vol. 530, no. 7589, p. 144, 2016.

[19] "Web search for a planet: The google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, 2003.

[20] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, 2013, pp. 315–323.

[21] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, vol. 162, pp. 83–112, 2017.

[22] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," *Advances in Neural Information Processing Systems 28*, 2014.

[23] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, 1970.

[24] C. Broyden, "The convergence of a class of double-rank minimization algorithms," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970.

[25] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, pp. 317–322, 1970.

[26] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Mathematics of computation*, vol. 24, no. 111, pp. 647–656, 1970.

[27] R. H. Byrd, J. Nocedal, and Y.-X. Yuan, "Global convergence of a class of quasi-Newton methods on convex problems," *SIAM Journal on Numerical Analysis*, no. 5, pp. 1171–1190, 1987.

[28] W. C. Davidon, "Variable metric method for minimization," Argonne National Laboratory, Tech. Rep. ANL-5990, 1959.

[29] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *The Computer Journal*, vol. 6, no. 2, pp. 163–168, 1963.

[30] R. Schnabel, "Quasi-Newton methods using multiple secant quotations," University of Colorado at Boulder, Tech. Rep. CU-CS-247-83, 1983.

[31] R. Gower, D. Goldfarb, and P. Richtárik, "Stochastic block BFGS : Squeezing more curvature out of data," vol. 48, pp. 1869–1878, 2016.

[32] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A stochastic quasi-newton method for large-scale optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1008–1031, 2016.

[33] Yu. Nesterov, *Introductory Lectures on Convex Optimization*. New York: Springer Science+Business Media, 2004.

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[35] D. Steinkraus, I. Buck, and P. Simard, "Using gpus for machine learning algorithms," in *International Conference on Document Analysis and Recognition*, 2005.

[36] T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," *CoRR*, vol. abs/1802.09941, 2018.

[37] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv:1706.02677*, 2017.

[38] S. Zhang, A. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *NIPS*, 2015.

[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[40] D. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[41] M. J. D. Powell, "Some global convergence properties of a variable metric algorithm for minimization without exact line searches," in *Nonlinear Programming*, R. Cottle and C. Lemke, Eds., vol. IX, SIAM-AMS Proceedings, 1976.

[42] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd. Cambridge: Cambridge University Press, 2013.

[43] J. E. Dennis Jr. and J. J. Moré, "Characterization of superlinear convergence and its application to quasi-Newton methods," *Mathematics of Computation*, vol. 28, no. 106, pp. 549–560, 1974.

[44] A. Griewank and P. L. Toint, "Local convergence analysis for partitioned quasi-Newton updates," *Numerical Mathematics*, vol. 39, pp. 429–448, 1982.

[45] D.-H. Li and M. Fukushima, "On the global convergence of the BFGS method for nonconvex unconstrained optimization problems," *SIAM Journal on Optimization*, vol. 11, no. 4, pp. 1054–1064, 2001.

[46] ——, "A modified BFGS method and its global convergence in nonconvex minimization," *Journal of Computational and Applied Mathematics*, vol. 129, pp. 15–35, 2001.

[47] M. J. D. Powell, "Algorithms for nonlinear constraints that use Lagrangian functions," *Mathematical Programming*, vol. 14, pp. 224–248, 1978.

[48] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201–213, 2 2002.

[49] M. Schmidt, "MinFunc: Unconstrained differentiable multivariate optimization in Matlab," `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`, 2005.

[50] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Models and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.

[51] M. J. Weinstein and A. V. Rao, "Adigator, a toolbox for the algorithmic differentiation of mathematical functions in MATLAB using source transformation via operator overloading," *ACM Transactions on Mathematical Software*, 2017.

[52] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, 2011.

[53] I. Maros and C. Mészáros, "A repository of convex quadratic programming problems," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 671–681, 1999.

[54] W. Hager, "A collection of optimization test problems," `http://users.clas.ufl.edu/hager/coap/Pages/matlabpage.html`, Accessed 2016-06-13.

[55] Yu. Nesterov and A. Nemirovski, *Interior-point polynomial algorithms in convex programming*. Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[56] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher, "Composite self-concordant minimization," *Journal of Machine Learning Research*, vol. 16, no. Mar, pp. 371–416, 2015.

[57] Y. Zhang and L. Xiao, "DiSCO: Distributed optimization for self-concordant empirical loss," in *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*, vol. 37, 2015, pp. 362–370.

[58] Z. Lu, "Randomized block proximal damped Newton methods for composite self-concordant minimization," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1910–1942, 2017.

[59] C. G. Broyden, "Quasi-Newton methods and their application to function minimisation," *Mathematics of Computation*, vol. 21, no. 99, pp. 368–381, 1967.

[60] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd. New York: Springer Science+Business Media, 2006.

[61] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Mathematics of Computation*, vol. 35, no. 151, pp. 773–782, 1980.

[62] C. Zhou, W. Gao, and D. Goldfarb, "Stochastic adaptive quasi-Newton methods for minimizing expected values," *Proceedings of the 34th ICML (PMLR)*, vol. 70, pp. 4150–4159, 2017.

[63] A. Rodomanov and D. Kropotov, "A superlinearly-convergent proximal Newton-type method for the optimization of finite sums," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 2597–2605.

[64] A. Gholami, A. Azad, P. Jin, K. Keutzer, and A. Buluc, "Integrated model, batch, and domain parallelism in training neural networks," *Proceedings of the 30th Syposium on Parallelism in Algorithms and Architectures*, pp. 77–86, 2018.

[65] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *NIPS*, 2011.

[66] P. Chaudhari, C. Baldassi, R. Zecchina, S. Soatto, A. Talwalkar, and A. Oberman, "Parle: Parallelizing stochastic gradient descent," in *arXiv:1707.00424*, 2017.

[67] X. Li, J. Lu, R. Arora, J. Haupt, H. Liu, Z. Wang, and T. Zhao, "Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization," *IEEE Transactions on Information Theory*, vol. PP, 2019.

[68] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, *et al.*, "Large scale distributed deep networks," in *NIPS*, 2012.

[69] A. Sergeev and M. D. Balso, "Horovod: Fast and easy distributed deep learning in TensorFlow," *CoRR*, vol. abs/1802.05799, 2018.

[70] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2017.

[71] S. L. Smith and Q. V. Le, "A bayesian perspective on generalization and stochastic gradient descent," *International Conference on Learning Representations*, 2018.

[72] S. Jastrzębski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, "Finding flatter minima with sgd," in *ICLR Workshop Track*, 2018.

[73] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, "Entropy-SGD: Biasing gradient descent into wide valleys," *International Conference on Learning Representations*, 2017.

[74] Y. You, I. Gitman, and B. Ginsburg, "Scaling SGD batch size to 32k for imagenet training," *International Conference on Learning Representations*, 2018.

[75] T. Sun, R. Hannah, and W. Yin, "Asynchronous coordinate descent under more realistic assumptions," *Advances in Neural Information Processing Systems 30*, pp. 6182–6190, 2017.

[76] Z. Peng, Y. Xu, M. Yan, and W. Yin, "On the convergence of asynchronous parallel iteration with arbitrary delays," *Journal of the Operations Research Society of China*, vol. 7, no. 1, pp. 5–42, 2019.

[77] S. Zhang, "Distributed stochastic optimization for deep learning," Ph.D. dissertation, New York University, 2016.

[78] J. Sun, Q. Qu, and J. Wright, "Complete dictionary recovery over the sphere i: Overview and the geometric picture," *IEEE Trans. Information Theory*, vol. 63, no. 2, pp. 853–884, 2017.

[79] ——, "Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method," *IEEE Trans. Information Theory*, vol. 63, no. 2, pp. 885–914, 2017.

[80] R. Ge, J. Lee, and T. Ma, "Matrix completion has no spurious local minimum," vol. 29, 2016.

[81] R. Ge, C. Jin, and Y. Zheng, "No spurious local minima in nonconvex low rank problems: A unified geometric analysis," vol. 70, 2017, pp. 1233–1242.

[82] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *AISTATS*, 2015.

[83] K. Kawaguchi, "Deep learning without poor local minima," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[84] A. Agarwal, M. Wainwright, P. Bartlett, and P. Ravikumar, "Information-theoretic lower bounds on the oracle complexity of convex optimization," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[85] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *Advances in Neural Information Processing Systems 27*, vol. 27, pp. 2933–2941, 2014.

[86] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[87] R. Glowinski and A. Marroco, "On the approximation by finite elements of order one, and resolution, penalisation-duality for class of nonlinear dirichlet problems," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 9, no. 2, pp. 41–76, 1975.

[88] D. Gabay, "Applications of the method of multipliers to variational inequalities," in *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*, M. Fortin and R. Glowinski, Eds., North-Holland, 1983, pp. 299–331.

[89] J. Eckstein and D. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 3, pp. 293–318, 1992.

[90] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, pp. 964–979, 1979.

[91] J. Eckstein and W. Yao, "Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives," *Pacific Journal on Optimization*, vol. 11, no. 4, pp. 619–644, 2015.

[92] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, vol. 155, pp. 57–79, 2016.

[93] T. Lin, S. Ma, and S. Zhang, "Global convergence of unmodified 3-block ADMM for a class of convex minimization problems," *Journal of Scientific Computing*, vol. 76, no. 1, pp. 69–88, 2018.

[94] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012.

[95] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the ADMM with multi-block variables," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1478–1497, 2015.

[96] ——, "On the sublinear convergence rate of multi-block ADMM," *Journal of the Operations Research Society of China*, vol. 3, no. 3, pp. 251–271, 2015.

[97] M. Li, D. Sun, and K.-C. Toh, "A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block," *Asia-Pacific Journal of Operational Research*, vol. 32, no. 04, p. 1 550 024, 2015.

[98] D. Davis and W. Yin, "A three-operator splitting scheme and its optimization applications," *Set-Valued and Variational Analysis*, vol. 25, no. 4, pp. 829–858, 2017.

[99] E. Ryu, "Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting," 2018, `https://arxiv.org/abs/1802.07534`.

[100] D. Sun, K. Toh, and L. Yang, "A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 882–915, 2015.

[101] X. Li, D. Sun, and K.-C. Toh, "A schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions," *Mathematical Programming*, vol. 155, pp. 333–373, 1-2 2016.

[102] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o(1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[103] L. Chen, D. Sun, and K.-C. Toh, "An efficient inexact symmetric gauss–seidel based majorized ADMM for high-dimensional convex composite conic programming," *Mathematical Programming*, vol. 161, no. 1-2, pp. 237–270, 2017.

[104] Z. Lin, R. Liu, and H. Li, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning," *Machine Learning*, vol. 99, no. 2, pp. 287–325, 2015.

[105] R. I. Boţ and D.-K. Nguyen, "The proximal alternating direction method of multipliers in the non-convex setting: Convergence analysis and rates," *arXiv:1801.01994*, 2018.

[106] J. J. Wang and W. Song, "An algorithm twisted from generalized ADMM for multi-block separable convex minimization models," *Journal of Computational and Applied Mathematics*, vol. 309, pp. 342–358, 2017.

[107] M. Sun and H. Sun, "Improved proximal ADMM with partially parallelsplitting for multi-block separable convex programming," *Journal of Applied Mathematics and Computing*, vol. 58, no. 1–2, pp. 151–181, 2018.

[108] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[109] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[110] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM Journal on Optimization*, vol. 25, pp. 2434–2460, 2015.

[111] J. Zhang, S. Ma, and S. Zhang, "Primal-dual optimization algorithms over riemannian manifolds: An iteration complexity analysis," 2017, `https://arxiv.org/abs/1710.02236`.

[112] B. Jiang, T. Lin, S. Ma, and S. Zhang, "Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis," *Computational Optimization and Applications*, vol. 72, no. 1, pp. 115–157, 2019.

[113] D. Hajinezhad, T.-H. Chang, X. Wan, Q. Shi, and M. Hong, "Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4742–4746, 2016.

[114] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein, "Training neural networks without gradients: A scalable ADMM approach," *Proceedings of the 33rd International Conference on Machine Learning (PMLR)*, vol. 48, pp. 2722–2731, 2016.

[115] J. Bolte, S. Sabach, and M. Teboulle, "Nonconvex lagrangian-based optimization: Monitoring schemes and global convergence," *Mathematics of Operations Research*, vol. 43, no. 4, pp. 1210–1232, 2018.

[116] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[117] ——, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[118] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[119] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, pp. 3736–3745, 12 2006.

[120] S. Wang, L. Zhang, Y. Liang, and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," *Computer Vision and Pattern Recognition*, 2012.

[121] M. Hubert and S. Engelen, "Robust pca and classification in biosciences," *Bioinformatics*, vol. 20, pp. 1728–1736, 11 2004.

[122] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, 3 2011.

[123] D. Driggs, S. Becker, and A. Aravkin, "Adapting regularized low-rank models for parallel architectures," 2017, `https://arxiv.org/abs/1702.02241`.

[124] S. Burer and R. D. C. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[125] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations, IBM Research Symposia Series*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds., Springer, 1972, pp. 85–103.

[126] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[127] X. Bai and K. Scheinberg, "Alternating direction methods for non convex optimization with applications to second-order least-squares and risk parity portfolio selection," Tech. Rep., 2015, `http://www.optimization-online.org/DB_HTML/2015/02/4776.html`.

[128] Yu. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[129] L. Chen, D. Sun, and K.-C. Toh, "A note on the convergence of ADMM for linearly constrained convex optimization problems," *Computational Optimization and Applications*, vol. 66, no. 2, pp. 327–343, 2017.

[130] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.

[131] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[132] Y. Cui, X. Li, D. Sun, and K.-C. Toh, "On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions," *Journal of Optimization THeory and Applications*, vol. 169, no. 3, pp. 1013–1041, 2016.

[133] M. Li, D. Sun, and K.-C. Toh, "A majorized ADMM with indefinite proximal terms for linearly constrained convex composite optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 922–950, 2016.

[134] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946–977, 2013.

[135] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, 1st, ser. Grundlehren der mathematischen Wissenschaften 317. Springer-Verlag Berlin Heidelberg, 1997.

[136] R. Janin, "Directional derivative of the marginal function in nonlinear programming," in *Sensitivity, Stability, and Parametric Analysis, Mathematical Programming Studies*, A. V. Fiacco, Ed., vol. 2, Springer, Berlin, Heidelberg, 1984.

[137] S. Lu, "Implications of the constant rank constraint qualification," *Mathematical Programming*, vol. 126, no. 2, pp. 365–392, 2011.

[138] A. Sokal, "A really simple elementary proof of the uniform boundedness theorem," *American Mathematical Monthly*, vol. 118, no. 5, pp. 450–452, May 2011.

[139] Yurii Nesterov and B. Polyak, "Cubic regularization of newton method and its global performance," *Mathematical Programming*, vol. 108, no. 1, pp. 177–205, 2006.

[140] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the

kurdyka-łojasiewicz inequality," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 438–457, 2010.

[141] B. Jiang, S. Ma, and S. Zhang, "Alternating direction method of multipliers for real and complex polynomial optimization models," *Optimization*, vol. 63, no. 6, pp. 883–898, 2014.

[142] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers of Mathematics in China*, vol. 7, no. 2, pp. 365–384, 2012.

[143] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.